ASIA PACIFIC UNIVERSITY
OF TECHNOLOGY & INNOVATION

Individual Assignment – Loan Approval Status Prediction

TECHNOLOGY PARK MALAYSIA

CT050-3-M-DAP

Data Analytical Programming

APUMF2310DSBA(DE)(PR)

Student's TP: TP078400

Student's Name: Mr. VICTOR HEW XIN KAI

Lecturer's Name: Mr. DHASON PADMAKUMAR

Table of Contents

1.0 Introduction

In this era of technological globalization, the importance of small and medium-sized enterprises (SMEs) is sometimes undermined compared to multinational conglomerates (MNCs) and other large enterprises in driving the nation's economy to greater heights due to differences in scale of operations. Unlike large enterprises, SMEs need to fulfil their financial appetite with sufficient funding from financing companies to boost their growth in terms of operational expansions, investments, and market competitiveness. Nevertheless, the loan approval process of financing companies may not keep up with their eagerness to help SMEs financially due to its complexity, inefficiency, and unreliability in ensuring the financial appetite of loan applicants with the most eligibility being fulfilled on time, which can stunt the expansion of promising SMEs. Another downside revolves around the decision-making of financial institutions which mistakenly approve ineligible loan applications, thus potentially leading to significant financial loss.

Therefore, a quick, streamlined and customer-centric loan approval predictive model can be developed by data scientists to address this pressing need to streamline the loan approval process using advanced data analytics and machine learning methods. By leveraging the analysis of historical loan application data, data scientists can build a predictive classification model that can accurately give the stamp of approval or rejection to the loan applicants based on different factors but are not limited to clientele's characteristics such as credit scores, loan amount and income. The prediction of loan approval can assist loan officers in the financing companies in accelerating their data-driven decision-making process for the provision of loan support to SMEs. As a result, SMEs can seize growth and investment opportunities at the right time without delays, thus fostering the relationship between SME loan applicants and financing companies. From the lenders' perspectives, the financing companies can decrease their operational costs and boost the performance of the loan portfolio, thus contributing to the economic and innovation landscape and financial service transformation from a broader perspective.

2.0 Background of Lasiandra Finance Inc. (LFI)

Headquartered in New York, LFI is one of the prominent figures with a great reputation in the financing industry. It provides customer-centric, personalized loan support to SMEs with specific business needs that require the additional financial push to realise their business potential through the expansion of their business landscape to a new scale. Over the past few years, the

business growth of LFI has significantly reached a new level so that they can reach out to more SMEs to help them financially, which is in alignment with their vision. To keep up with the increasing business growth accompanied by the increasing number of loan applicants, LFI needs to up their financing game in terms of the efficiency and accuracy of the existing loan approval process to stay competitive among its competitors. The existing loan approval process at LFI is managed through a multistep, manual authentication and evaluation system to make sure that the SME applicants are eligible for funding based on different factors including but not limited to the demographics of the SME owner, income and loan amount and history. The step-by-step end-to-end loan process is outlined below:

1. Online or physical submission of loan applications by SMEs, detailing SME information such as demographics, credit history, collateral details, business blueprint and other financial details related to the SME
2. Screening loan applications by loan officers for document completeness and verification
3. Evaluation of applicants' creditworthiness based on factors like cash flow, income and loan history
4. Evaluation of the SME business of applicants to better understand their business positioning and growth prospect
5. Evaluation of potential loan approval risk by loan officers
6. Communication of final loan disbursement decision
7. Discussion and signing of loan agreement and loan disbursement

However, maintaining the comprehensiveness of the loan approval process comes with the consumption of consistencies, resources, and time as the number of applicants grows over time. LFI wants clarity on their decision-making related to the worthiness of the approved loan applicant. Therefore, automation is the key towards the streamlining of the complicated loan approval process. Recognizing this, LFI has tasked one of the recently hired data scientists to develop and implement the most accurate and efficient loan approval predictive model based on the comprehensive analysis of the historical loan applicant data.

3.0 Assumption & Justification

Several assumptions are made by the data scientist. The first assumption revolves around data complexity as the given historical loan applicant dataset comprises different quantitative and qualitative variables such as the demographics of SME loan applicants, income as well as loan amount, duration, and history, indicating high data dimensionality. The second assumption is that LFI is obligated to comply with the regulatory requirements of New York as they are in the financing industry which is more sensitive than other industries in terms of data privacy. The third assumption is that the loan approval process of LFI needs to cope with the large volume of loan applications as they cater to more SME customers while ensuring the high reliability and accuracy of the loan approval predictive model to reduce business risks. Therefore, the statistical analysis system (SAS) is chosen as the go-to data analytical tool which supports the end-to-end data analytical workflow starting from the sample and explore to the modify and model stages.

One of the justifications for choosing SAS revolves around data management and analytics. In SAS, the data scientist can manipulate the large volume of loan application data at their disposal such as data cleaning, transforming and imputation of missing values as well as performing data visualization for exploratory data analysis so that the subsequent data analysis can benefit from the cleaned, high-quality data and the model performance can be boosted using those data. Another justification is the remarkable capability of automated predictive modelling as the tasked data scientist can utilize a wide range of machine learning algorithms such as logistic regression, decision tree and random forest to compare the performance of the loan approval predictive model across different algorithms and ultimately build the most reliable model that is scalable enough to manage more loan applications in an automated, seamless manner as the business grows over time. The third justification relates to data security and compliance, which is one of the advantages of SAS over other programming languages. Loan application data are highly sensitive as they involve confidential business information. If leaked, the SME will be disadvantaged compared to competitors who may be exposed to those leaked data. The high-security nature of SAS can prevent this from happening, thus boosting the protection of sensitive loan application data.

4.0 Data Dictionary/Metadata

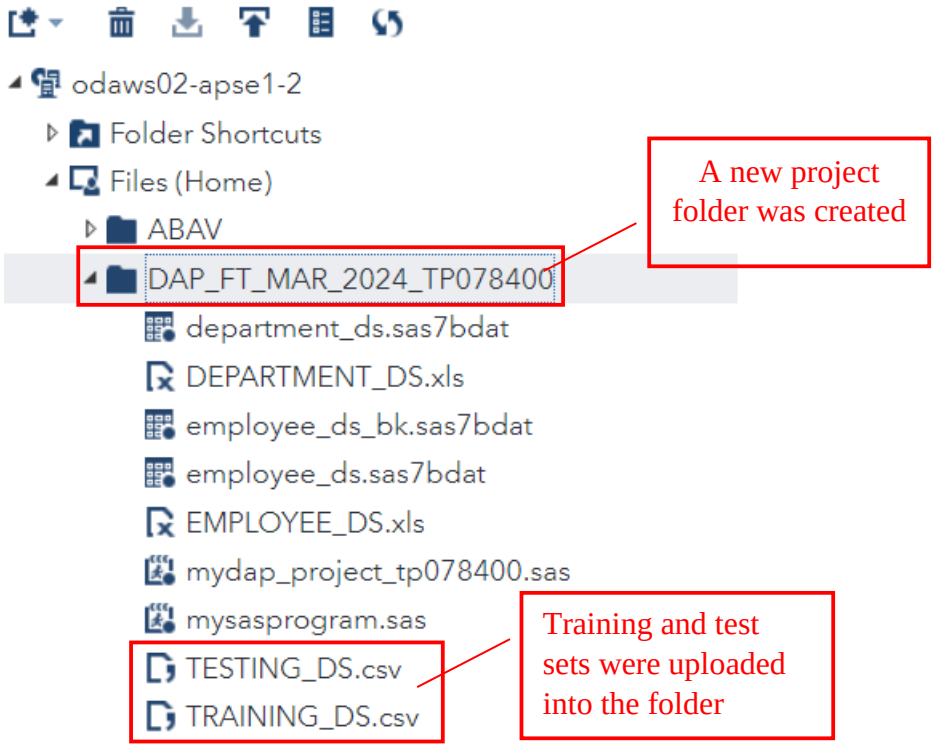
4.1 Details of the Data Dictionary/Metadata

Variables	Description	Data Type	Length	Sample Data
SME_LOAN_ID_NO	The unique identification number of SME loan applications	Char	8	LP001002 LP002990 LP001050 LP001047 LP001086
GENDER	The gender of the loan applicants (Male/Female)	Varchar	6	Female Male Male Female Male
MARITAL_STATUS	The marital status of the loan applicants (Married/Not Married)	Varchar	11	Not Married Married Not Married Married Not Married
FAMILY_MEMBERS	Number of family members by loan applicants (0/1/2/3+)	Varchar	2	0 1 2 3+
LOAN_AMOUNT	Applied loan amount (in thousands) by loan applicants	Integer	5	349 110 35 133
QUALIFICATION	Education qualification of loan applicants	Varchar	14	Graduate Under Graduate Under Graduate Graduate
EMPLOYMENT	Employment status of loan applicants	Varchar	3	Yes No

CANDIDATE_INCOME	Monthly income of loan candidates	Integer	5	6000 3717 10750 2071 14999
GUARANTEE_INCOME	Monthly joint income of loan applicants	Integer	5	1430 0 11300 2583 983
LOAN_DURATION	Loan repayment duration	Integer	3	360 180 480 12 36
LOAN_HISTORY	Historical loan records (Positive = 1/Negative = 0)	Integer	1	1 0
LOAN_LOCATION	Location of loan applicants (City/Town/Village)	Varchar	7	City Town Village
LOAN_APPROVAL_STATUS	Loan Approval (Y = Yes/N = No)	Char	1	Y N

4.2 Upload the Datasets to the Project Folder on SAS

4.2.1 Screenshots/Outputs

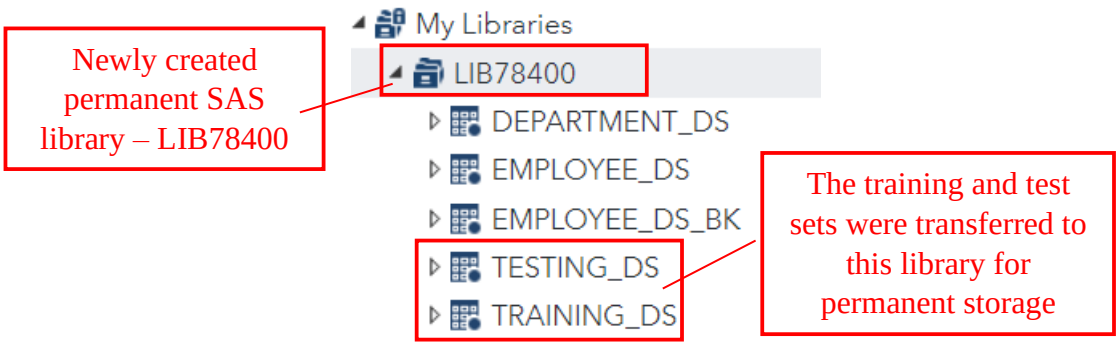


4.2.2 Descriptions

In SAS Studio, a new folder named DAP_FT_MAR_2024_TP078400 was created by the data scientist and two CSV-format loan approval datasets, namely TRAINING_DS and TESTING_DS, both of which represented the training and test sets respectively, were then uploaded into the folder.

4.3 Transfer the Datasets from the Project Folder to the Newly Created Permanent SAS Library – LIB78400

4.3.1 Screenshots/Outputs



4.3.2 Descriptions

A new library named LIB78400 was created by the data scientist and two CSV-format loan approval datasets, namely TRAINING_DS and TESTING_DS, both of which represented the training and test sets respectively and were uploaded to the DAP_FT_MAR_2024_TP078400 folder earlier, were then transferred to this library for permanent storage.

4.4 Display the Dataset Structure – LIB78400.TRAINING_DS (Method 1)

4.4.1 SAS Codes

```
/*********************************************************
Developer name: Mr Victor Hew Xin Kai
Job position: Data Scientist, APU SDN BHD
Program name: mydap_project_tp078400.sas
Description: Loan application status prediction
Date first written: Fri,26-Apr-2024
Date last updated: Fri,26-Apr-2024
Folder name:  DAP_FT_MAR_2024_TP078400
Library name:  LIB78400
*********************************************************/

/*SAS Codes to display the data dictionary of LIB78400.TRAINING_DS */

PROC SQL;

DESCRIBE TABLE LIB78400.TRAINING_DS;

RUN;
```

4.4.2 Screenshot(s)/Output(s)

```
create table LIB78400.TRAINING_DS( bufsize=131072 )
  (
   SME_LOAN_ID_NO char(8) format=$8. informat=$8.,
   GENDER char(6) format=$6. informat=$6.,
   MARITAL_STATUS char(11) format=$11. informat=$11.,
   FAMILY_MEMBERS char(2) format=$2. informat=$2.,
   QUALIFICATION char(14) format=$14. informat=$14.,
   EMPLOYMENT char(3) format=$3. informat=$3.,
   CANDIDATE_INCOME num format=BEST12. informat=BEST32.,
   GUARANTEE_INCOME num format=BEST12. informat=BEST32.,
   LOAN_AMOUNT num format=BEST12. informat=BEST32.,
   LOAN_DURATION num format=BEST12. informat=BEST32.,
   LOAN_HISTORY num format=BEST12. informat=BEST32.,
   LOAN_LOCATION char(7) format=$7. informat=$7.,
   LOAN_APPROVAL_STATUS char(1) format=$1. informat=$1.
  );
```

4.4.3 Description

The data scientist used the DESCRIBE TABLE statement in PROC SQL to show the description of the table structure of LIB78400.TRAINING_DS which was stored permanently in the LIB78400 library earlier as shown in the output above.

4.5 Display the Dataset Structure – LIB78400.TRAINING_DS (Method 2)

4.5.1 SAS Codes

```
20 PROC CONTENTS DATA = LIB78400.TRAINING_DS;
21
22 RUN;
```

4.5.2 Screenshot(s)/Output(s)

The CONTENTS Procedure

Data Set Name	LIB78400.TRAINING_DS	Observations	614
Member Type	DATA	Variables	13
Engine	V9	Indexes	0
Created	27/05/2024 21:46:07	Observation Length	96
Last Modified	27/05/2024 21:46:07	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	SOLARIS_X86_64, LINUX_X86_64, ALPHA_TRU64, LINUX_IA64		
Encoding	utf-8 Unicode (UTF-8)		

Engine/Host Dependent Information	
Data Set Page Size	131072
Number of Data Set Pages	1
First Data Page	1
Max Obs per Page	1363
Obs in First Data Page	614
Number of Data Set Repairs	0
Filename	/home/u63691887/DAP_FT_MAR_2024_TP078400/training_ds.sas7bdat
Release Created	9.0401M7
Host Created	Linux
Inode Number	3087856886
Access Permission	rw-r--r--
Owner Name	u63691887
File Size	256KB
File Size (bytes)	262144

Alphabetic List of Variables and Attributes					
#	Variable	Type	Len	Format	Informat
7	CANDIDATE_INCOME	Num	8	BEST12.	BEST32.
6	EMPLOYMENT	Char	3	$3.	$3.
4	FAMILY_MEMBERS	Char	2	$2.	$2.
2	GENDER	Char	6	$6.	$6.
8	GUARANTEE_INCOME	Num	8	BEST12.	BEST32.
9	LOAN_AMOUNT	Num	8	BEST12.	BEST32.
13	LOAN_APPROVAL_STATUS	Char	1	$1.	$1.
10	LOAN_DURATION	Num	8	BEST12.	BEST32.
11	LOAN_HISTORY	Num	8	BEST12.	BEST32.
12	LOAN_LOCATION	Char	7	$7.	$7.
3	MARITAL_STATUS	Char	11	$11.	$11.
5	QUALIFICATION	Char	14	$14.	$14.
1	SME_LOAN_ID_NO	Char	8	$8.	$8.

4.5.3 Description

The data scientist used PROC CONTENTS to display the information of the LIB78400.TRAINING_DS dataset as shown in the outputs above aside from using the DESCRIBE TABLE statement in PROC SQL.

5.0 Literature Review

5.1 Loan Approval Process

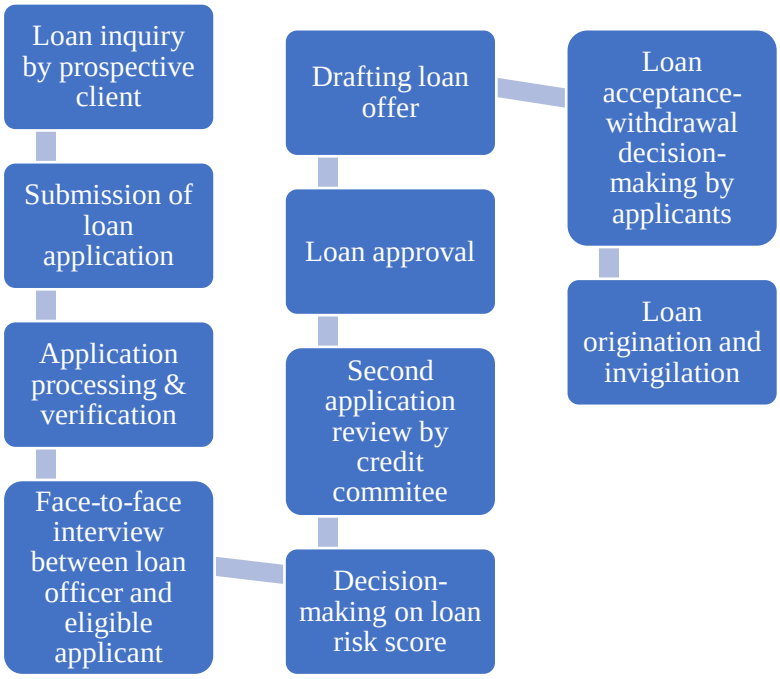
Undoubtedly, the loan approval process is one of the core components for financing companies like LFI which prides itself on catering for the funding needs of LFI. This process is set in place to protect the interest of financing companies in ensuring that the loanees can repay within the said duration and manage loan risk expectations. Some past research such as Agarwal and Ben-David (2018) and Campbell et al. (2019) cast a lens on the specifics of the typical traditional loan approval process in financial institutions.

First, the inquiries about potential business loans were made by prospective SME clients to the loan officers who typically suggest clients submit loan applications together with the required details and supporting documents such as proposed loan amount, loan repayment duration, financial statements, tax records and more. After the application submission, loan officers will process them through additional hard information gathering like credit ratings and verification for creditworthiness assessment. They will ultimately drop ineligible applicants with low credit scores and above-average debt-to-income ratios.

Initial eligible applicants are then subjected to a face-to-face interview with loan officers. During the interview, applicants will do their storytelling on personal financial circumstances and the reasons behind the degradation of credit scores so that loan officers can build rapport with them and better understand the applicants' behaviours and financial accountability as a whole before using the hard and soft details collected to come up with internal loan risk scores. Although there is the establishment of loan guidelines, it is up to the disposal of officers to whether adhere to or deviate from them as soft information and subjective judgement may carry greater weight in their risk decision-making depending on their work personality, which may be detrimental to the reputation of the financial institution due to potential drop in loan quality (Campbell et al., 2019). The second line of loan decision-making defence is the branch manager and other loan officers who perform application reviews and lower the risk rating threshold on a case-by-case basis.

If the application is approved, a loan offer letter with the proposed terms and conditions is prepared and shared with the loan applicants who can make loan term negotiations with the

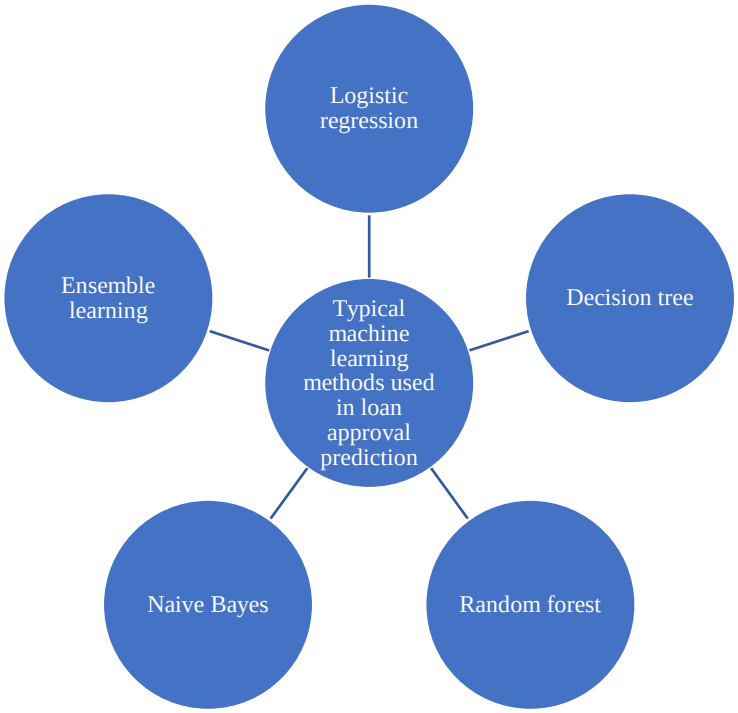
credit committee before proceeding to the acceptance-withdrawal decisions. The financing institution originates the accepted loan and then puts it into its financial records and invigilates the loan throughout its life, with annual face-to-face discussions being held between the applicant and the loan officer if they need more loan support. The summary of the flow of the traditional loan approval process is visualized below (Agarwal and Ben-David, 2018; Campbell et al., 2019):



However, there are many manual steps involved here indicating that the processing time for loan application is lengthy. This calls for the need to automate the loan approval process using predictive modelling, which streamlines the end-to-end process and reduces human bias in loan decision-making.

5.2 Predictive Modeling for Loan Approval Process

Based on historical loan application data, financing companies can leverage the power of machine learning algorithms in predicting whether the loan application is approved or rejected as well as understanding the most important variables in loan approval. Some of the typical machine learning methods used in this context based on past studies are visualized below using a tree map diagram:



Nureni and Adekola (2022) compared the usage of different supervised and unsupervised machine learning algorithms such as linear and logistic regression, support vector machine (SVM), K-Nearest Neighbour (KNN), K-Means clustering, and tree-based algorithms including decision tree and random forest in loan approval system in the banking industry. The results suggested that the worst model performance belonged to K-Means and KNN in terms of accuracy, F1-score, precision and recall. The top four models which delivered the best performance regardless of rankings belonged to random forest, naïve Bayes, and logistic and linear regression, with the ensemble learning method being one of the suggested ways to further boost the performances in future studies. Bansode et al. (2022) further extended the results by including tree-based algorithms, logistic regression, SVM and KNN to forecast loan approval as a binary classification problem with a 70-30 train-test split. The findings indicated that logistic regression produced the greatest accuracy of 84.38%. Some of the most important variables in predicting loan approval are credit history and income, indicating that loan applicants with positive past loan records, high income and loan amount with decreased interest rate would increase their likelihood of having their loan request approved. This was supported by Sheikh et al. (2020) and Gopinath et al. (2021) who used logistic regression to build the loan approval predictive model and found that it delivered an above-average accuracy of 81.1% and 80.95%

respectively. The most important variables found by Sheikh et al. (2020) were almost the same as the findings from Bansode et al. (2022), which were credit score, income and loan amount.

Aside from logistic regression, random forest, a collection of decision trees, was also one of the most popular machine learning methods to be used in loan approval prediction. Zhu et al. (2019) proposed a machine learning-based solution using random forest as the foundation for developing the loan default forecasting model to reduce the loan default risk in a peer-to-peer platform. Using loan data with more than 115,000 observations along with 102 variables from a US-based financial institution, Lending Club, they found that random forest delivered the best model performance compared to logistic regression, SVM and decision tree. Orji et al. (2022) supported this result as they found that random forest predicted loan approval with an accuracy as high as 95.55%. Madaan et al. (2021) compared the model performance of tree-based methods such as random forest and decision tree in loan default forecasting using Lending Club's data and found that the accuracy of random forest was higher than decision tree, which was also supported by Bhargav and Sashirekha (2023). However, Tejaswini et al. (2020) suggested the opposite result that the decision tree was better than logistic regression and random forest.

Overall, predictive modelling facilitates the decision-making process of loan committees in the financial institution in approving or rejecting loan applications.

5.3 Customer Portfolio & Credit Risk Management

It is undeniable that analysing customer portfolios and managing credit risk is important so that the loan approval predictive models can take every pivotal factor into account, thus bringing greater insights into the creditworthiness of applicants in a holistic manner. Some core elements make up a comprehensive customer portfolio for loan applications which facilitate financial institutions' assessment of loan eligibility. For example, demographic information which comprises age, gender, marital status, loan purpose and others can indicate whether the loan applicant is a financially reliable borrower (Sun et al., 2023; Yao et al., 2019). Besides, financial information such as applicants' income, credit ratings and loan repayment history can also contribute to the analysis of loan customer portfolios (Broby, 2021; Kgoroeadira et al., 2019). The third element revolves around business details such as cash flow, revenue, profit and type of business.

Undoubtedly, there are significant relationships between the elements above and loan approval. For example, Kgoroeadira et al. (2019) found that employment, credit assessment, and housing were some of the demographic and financial factors of loan approval. This indicated that the likelihood of business peer-to-peer loan approval could decrease if the borrower were a self-employed renter with a low credit score. Likewise, if the applicants with high credit ratings were employed and bought a house, the loan approval likelihood would increase. Johnson et al. (2023) found that the loan amount by borrowers would increase if they requested monthly payments of business loans with below-average cost due to perceived total cost, while loan amount would decrease if budget-minded borrowers asked for monthly payments of business loans accompanied by above-average cost.

There are several ways to manage loan approval risk. The first one revolves around bolstering the standards of underwriting for credit risk management so that loan applicants with above-average creditworthiness would benefit from the stringent process of loan approval (Chen, 2024). The second way revolves around collateral where the risk of financial institutions as lenders can be reduced by including loan-securing assets as security if the borrower is unable to repay the loan amount within the certain timeframe (Heller et al., 2024). The third way revolves around leveraging machine learning techniques to predict loan default through the monitoring of the finance of loan applicants (Chudappa et al., 2023).

5.4 Bad Debts & Non-Performing Loans

The lifecycle of loan process comes to an end with the management of bad debts and non-performing loans (NPLs). NPLs are defined as default loans which are not paid by borrowers within the scheduled period. These types of loans are bad debts as it is near impossible to get back the repayments of default loans, so it is detrimental to the profitability and operational resources and functions of financial institutions. Cetin (2019) found that there was a significant negative relationship between NPLs and return on asset across European banks.

Therefore, several strategies must be implemented to manage these bad debts. Volkova et al. (2019) indicated that one of the ways can be liquidating collateral put up by borrowers. Miller (2023) indicated that flexible and diverse loan repayment plans can be introduced to borrowers who are unable to repay those loans on time due to certain business circumstances. In terms of flexibility and diversity, allowing partial loan repayment of different percentages can indicate to

the borrowers that lenders are eager to continue the lender-borrower relationship while providing more convenience to them and acknowledging their current business hardships, thus motivating them to perform repayment as soon as possible and the probability of payment recovery increases.

6.0 Analysis of the Variables in the Dataset – LIB78400.TRAINING_DS

6.1 Univariate Analysis of the Categorical Variables in the Dataset

6.1.1 MARITAL_STATUS

6.1.1.1 SAS Codes

```
30 TITLE 'Univariate Analysis of the Categorical Variable: MARITAL_STATUS';
31
32 PROC FREQ DATA = LIB78400.TRAINING_DS;
33
34 TABLE MARITAL_STATUS;
35
36 RUN;
37
38 ODS GRAPHICS / RESET WIDTH = 3.0 IN HEIGHT = 4.0 IN IMAGEMAP;
39
40 PROC SGPLOT DATA = LIB78400.TRAINING_DS;
41
42 VBAR MARITAL_STATUS;
43
44 TITLE 'Univariate Analysis of the Categorical Variable: MARITAL_STATUS';
45
46 RUN;
```

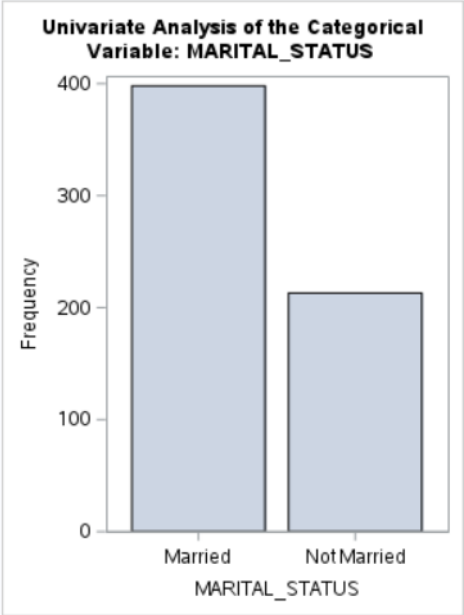
6.1.1.2 Screenshot(s)/Output(s)

Univariate Analysis of the Categorical Variable: MARITAL_STATUS

The FREQ Procedure

MARITAL_STATUS	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Married	398	65.14	398	65.14
Not Married	213	34.86	611	100.00
Frequency Missing = 3				

3 values were missing for MARITAL_STATUS



6.1.1.3 Description

The data scientist found that 65.14% of loan applicants were married (n = 398) and 34.86% of them were not married (n = 213), with three values missing in this MARITAL_STATUS variable. This indicated that there were more married than non-married loan applicants.

6.1.2 GENDER

6.1.2.1 SAS Codes

```
50 TITLE 'Univariate Analysis of the Categorical Variable: GENDER';
51
52 PROC FREQ DATA = LIB78400.TRAINING_DS;
53
54 TABLE GENDER;
55
56 RUN;
57
58 ODS GRAPHICS / RESET WIDTH = 3.0 IN HEIGHT = 4.0 IN IMAGEMAP;
59
60 PROC SGPLOT DATA = LIB78400.TRAINING_DS;
61
62 VBAR GENDER;
63
64 TITLE 'Univariate Analysis of the Categorical Variable: GENDER';
65
66 RUN;
```

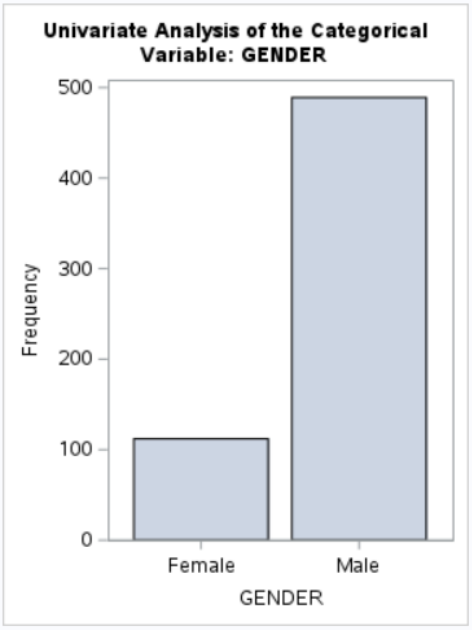
6.1.2.2 Screenshot(s)/Output(s)

Univariate Analysis of the Categorical Variable: GENDER

The FREQ Procedure

GENDER	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Female	112	18.64	112	18.64
Male	489	81.36	601	100.00
Frequency Missing = 13				

13 values were missing for GENDER



6.1.2.3 Description

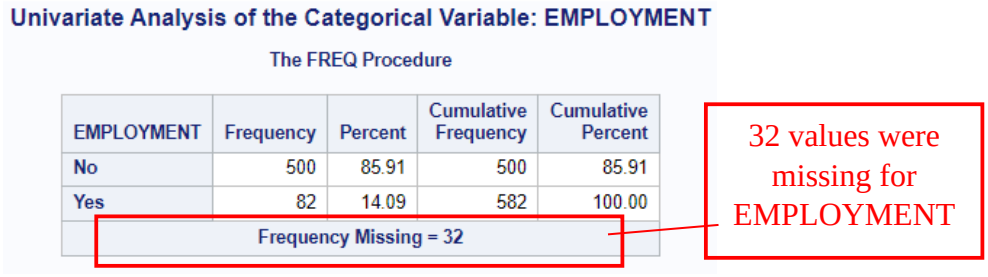
The data scientist found that 81.36% of loan applicants were male (n = 489) and 18.64% of them were female (n = 112), with 13 values missing in this GENDER variable. This indicated that there were more male than female applicants.

6.1.3 EMPLOYMENT

6.1.3.1 SAS Codes

```
90  TITLE 'Univariate Analysis of the Categorical Variable: EMPLOYMENT';
91
92  PROC FREQ DATA = LIB78400.TRAINING_DS;
93
94  TABLE EMPLOYMENT;
95
96  RUN;
97
98  ODS GRAPHICS / RESET WIDTH = 3.0 IN HEIGHT = 4.0 IN IMAGEMAP;
99
100 PROC SGPLOT DATA = LIB78400.TRAINING_DS;
101
102 VBAR EMPLOYMENT;
103
104 TITLE 'Univariate Analysis of the Categorical Variable: EMPLOYMENT';
105
106 RUN;
```

6.1.3.2 Screenshot(s)/Output(s)

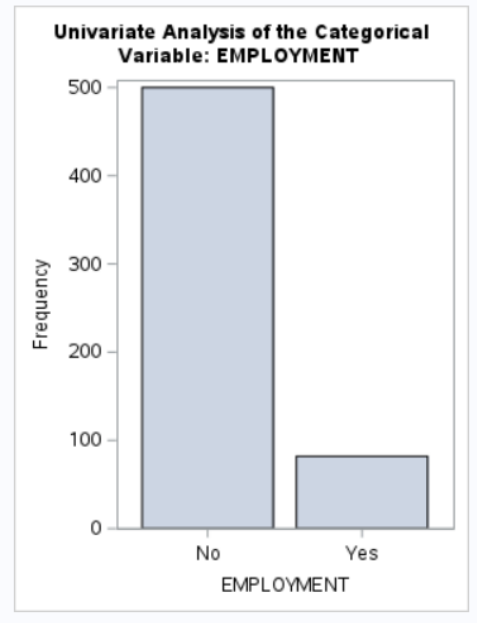


Univariate Analysis of the Categorical Variable: EMPLOYMENT

The FREQ Procedure

EMPLOYMENT	Frequency	Percent	Cumulative Frequency	Cumulative Percent
No	500	85.91	500	85.91
Yes	82	14.09	582	100.00
Frequency Missing = 32				

32 values were missing for EMPLOYMENT



6.1.3.3 Description

The data scientist found that 85.91% of loan applicants were employed (n = 500) and 14.09% of them were not employed (n = 82), with 32 values missing in this EMPLOYMENT variable. This indicated that there were more employed than non-employed loan applicants.

6.1.4 LOAN_APPROVAL_STATUS

6.1.4.1 SAS Codes

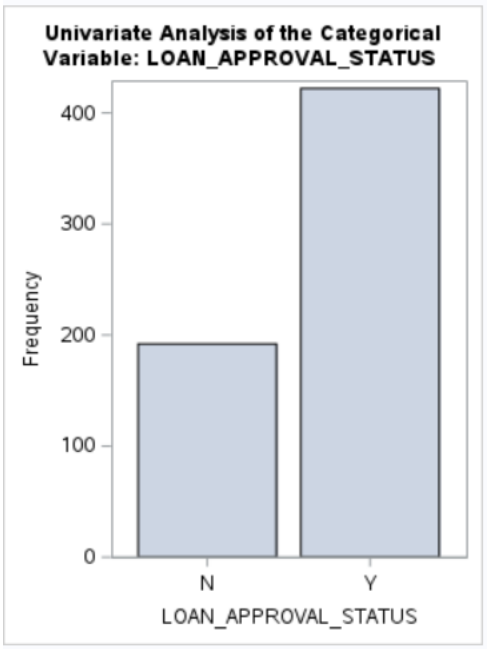
```
110 TITLE 'Univariate Analysis of the Categorical Variable: LOAN_APPROVAL_STATUS';
111
112 PROC FREQ DATA = LIB78400.TRAINING_DS;
113
114 TABLE LOAN_APPROVAL_STATUS;
115
116 RUN;
117
118 ODS GRAPHICS / RESET WIDTH = 3.0 IN HEIGHT = 4.0 IN IMAGEMAP;
119
120 PROC SGPLOT DATA = LIB78400.TRAINING_DS;
121
122 VBAR LOAN_APPROVAL_STATUS;
123
124 TITLE 'Univariate Analysis of the Categorical Variable: LOAN_APPROVAL_STATUS';
```

6.1.4.2 Screenshot(s)/Output(s)

Univariate Analysis of the Categorical Variable: LOAN_APPROVAL_STATUS

The FREQ Procedure

LOAN_APPROVAL_STATUS	Frequency	Percent	Cumulative Frequency	Cumulative Percent
N	192	31.27	192	31.27
Y	422	68.73	614	100.00



6.1.4.3 Description

The data scientist found that 68.73% of loan applicants had obtained loan approval (n = 422) and 31.27% of them were rejected from their loan applications (n = 192), with no values missing in this LOAN_APPROVAL_STATUS variable. This indicated that more loan applicants had their loan applications approved than rejected applicants.

6.2 Univariate Analysis of the Continuous Variables in the Dataset

6.2.1 LOAN_AMOUNT

6.2.1.1 SAS Codes

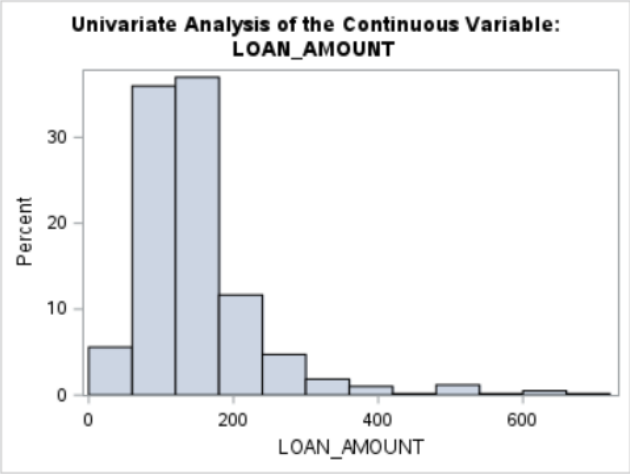
```
194 PROC MEANS DATA = LIB78400.TRAINING_DS N NMISS MIN MAX MEAN MEDIAN STD;
195
196 VAR LOAN_AMOUNT;
197
198 RUN;
199
200 ODS GRAPHICS / RESET WIDTH = 4.0 IN HEIGHT = 3.0 IN IMAGEMAP;
201
202 PROC SGPLOT DATA = LIB78400.TRAINING_DS;
203
204 HISTOGRAM LOAN_AMOUNT;
205
206 TITLE 'Univariate Analysis of the Continuous Variable: LOAN_AMOUNT';
207
208 RUN;
```

6.2.1.2 Screenshot(s)/Output(s)

Univariate Analysis of the Continuous Variable: LOAN_AMOUNT

The MEANS Procedure

Analysis Variable : LOAN_AMOUNT						
N	N Miss	Minimum	Maximum	Mean	Median	Std Dev
592	22	9.0000000	700.0000000	146.4121622	128.0000000	85.5873252



6.2.1.3 Description

The data scientist found that the typical and average loan amounts of all applicants were $128,000 and $146,412 respectively, with 22 values missing in this LOAN_AMOUNT variable. 36.99% and 35.98% of loan applicants applied for loan amounts of $150,000 and $90,000 respectively, both of which jointly represented the majority.

6.2.2 LOAN_DURATION

6.2.2.1 SAS Codes

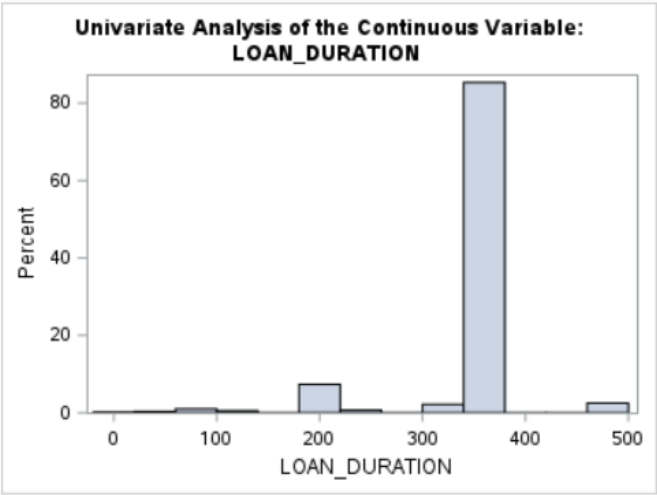
```
212 TITLE 'Univariate Analysis of the Continuous Variable: LOAN_DURATION';
213
214 PROC MEANS DATA = LIB78400.TRAINING_DS N NMISS MIN MAX MEAN MEDIAN STD;
215
216 VAR LOAN_DURATION;
217
218 RUN;
219
220 ODS GRAPHICS / RESET WIDTH = 4.0 IN HEIGHT = 3.0 IN IMAGEMAP;
221
222 PROC SGPLOT DATA = LIB78400.TRAINING_DS;
223
224 HISTOGRAM LOAN_DURATION;
225
226 TITLE 'Univariate Analysis of the Continuous Variable: LOAN_DURATION';
227
228 RUN;
```

6.2.2.2 Screenshot(s)/Output(s)

Univariate Analysis of the Continuous Variable: LOAN_DURATION

The MEANS Procedure

Analysis Variable : LOAN_DURATION						
N	N Miss	Minimum	Maximum	Mean	Median	Std Dev
600	14	12.0000000	480.0000000	342.0000000	360.0000000	65.1204099



6.2.2.3 Description

The data scientist found that the typical loan repayment duration of all applicants was 360 months (30 years), which represented 85.33% of the loan applicant population. 14 values were missing in this LOAN_DURATION variable.

6.2.3 CANDIDATE_INCOME

6.2.3.1 SAS Codes

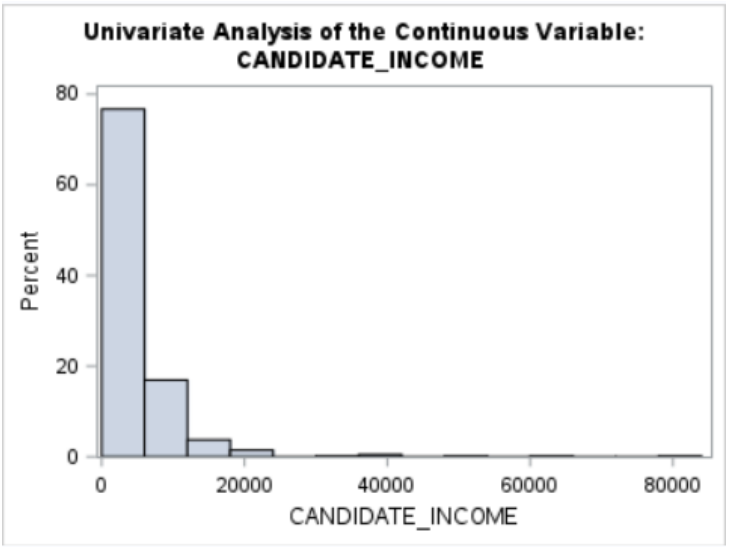
```
TITLE 'Univariate Analysis of the Continuous Variable: CANDIDATE_INCOME';

PROC MEANS DATA = LIB78400.TRAINING_DS N NMISS MIN MAX MEAN MEDIAN STD;

VAR CANDIDATE_INCOME;

RUN;

ODS GRAPHICS / RESET WIDTH = 4.0 IN HEIGHT = 3.0 IN IMAGEMAP;

PROC SGPLOT DATA = LIB78400.TRAINING_DS;

HISTOGRAM CANDIDATE_INCOME;

TITLE 'Univariate Analysis of the Continuous Variable: CANDIDATE_INCOME';

RUN;
```

6.2.3.2 Screenshot(s)/Output(s)

Univariate Analysis of the Continuous Variable: CANDIDATE_INCOME

The MEANS Procedure

Analysis Variable : CANDIDATE_INCOME						
N	N Miss	Minimum	Maximum	Mean	Median	Std Dev
614	0	150.0000000	81000.00	5403.46	3812.50	6109.04



6.2.3.3 Description

The data scientist found that the typical monthly income of all loan applicants was $3,812.50. 76.71% of applicants had a monthly income of $3,000. No missing values were found in this CANDIDATE_INCOME variable.

6.2.4 GUARANTEE_INCOME

6.2.4.1 SAS Codes

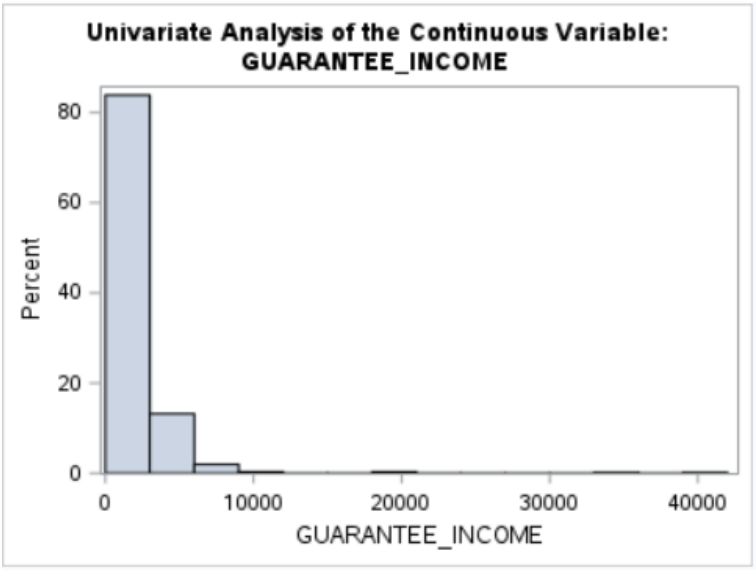
```
252 TITLE 'Univariate Analysis of the Continuous Variable: GUARANTEE_INCOME';
253
254 PROC MEANS DATA = LIB78400.TRAINING_DS N NMISS MIN MAX MEAN MEDIAN STD;
255
256 VAR GUARANTEE_INCOME;
257
258 RUN;
259
260 ODS GRAPHICS / RESET WIDTH = 4.0 IN HEIGHT = 3.0 IN IMAGEMAP;
261
262 PROC SGPLOT DATA = LIB78400.TRAINING_DS;
263
264 HISTOGRAM GUARANTEE_INCOME;
265
266 TITLE 'Univariate Analysis of the Continuous Variable: GUARANTEE_INCOME';
267
268 RUN;
```

6.2.4.2 Screenshot(s)/Output(s)

Univariate Analysis of the Continuous Variable: GUARANTEE_INCOME

The MEANS Procedure

Analysis Variable : GUARANTEE_INCOME						
N	N Miss	Minimum	Maximum	Mean	Median	Std Dev
614	0	0	41667.00	1621.25	1188.50	2926.25



6.2.4.3 Description

The data scientist found that the typical joint income of loan applicants was $1,158.50. 83.88% of applicants had a monthly joint income of $1,500. No missing values were found in this GUARANTEE_INCOME variable.

6.3 Bivariate Analysis of the Variables in the Dataset

6.3.1 Bivariate Analysis of the Variables (Categorical VS Categorical)

6.3.1.1 Bivariate Analysis of the Variables (Categorical – GENDER VS Categorical – LOAN_APPROVAL_STATUS)

6.3.1.1.1 SAS Codes

```
274 TITLE1 'Bivariate Analysis of the Variables';
275 TITLE2 'Categorical Variable - GENDER VS Categorical Variable - EMPLOYMENT';
276
277 PROC FREQ DATA = LIB78400.TRAINING_DS;
278
279 TABLE GENDER * EMPLOYMENT /
280 PLOTS = FREQPLOT (TWOWAY = STACKED SCALE = GROUPPCT);
281 RUN;
```

6.3.1.1.2 Screenshot(s)/Output(s)

Bivariate Analysis of the Variables
Categorical Variable - GENDER VS Categorical Variable - EMPLOYMENT

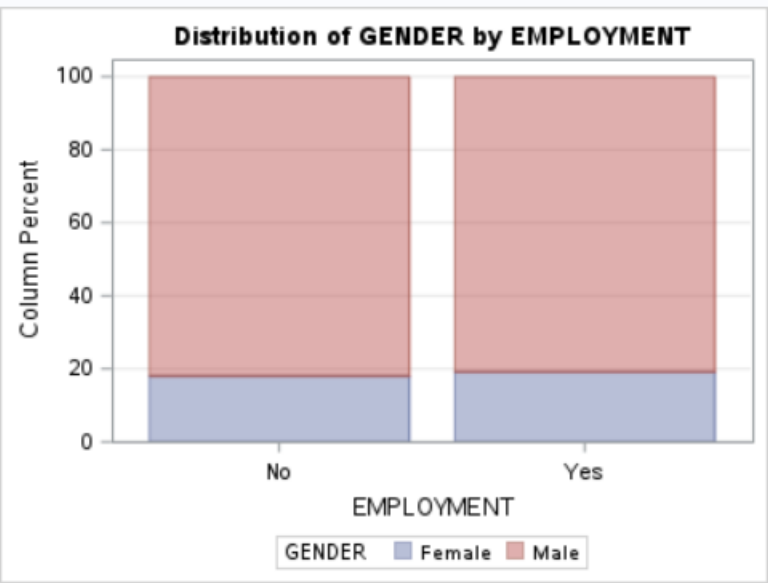
The FREQ Procedure

Frequency
Percent
Row Pct
Col Pct

Table of GENDER by EMPLOYMENT

GENDER	EMPLOYMENT		
	No	Yes	Total
Female	89 15.64 85.58 18.13	15 2.64 14.42 19.23	104 18.28
Male	402 70.65 86.45 81.87	63 11.07 13.55 80.77	465 81.72
Total	491 86.29	78 13.71	569 100.00

Frequency Missing = 45



6.3.1.1.3 Description

The data scientist found that for applicants with no employment, 70.65% of them were males (n = 402) and 15.64% were females (n = 89). For applicants with employment, 11.07% of them were males (n = 63) and 2.64% were females (n = 15). 45 missing values were found for the bivariate analysis of gender versus employment.

6.3.1.2 Bivariate Analysis of the Variables (Categorical – QUALIFICATION VS Categorical – LOAN_HISTORY)

6.3.1.2.1 SAS Codes

```
285 TITLE1 'Bivariate Analysis of the Variables';
286 TITLE2 'Categorical Variable - QUALIFICATION VS Categorical Variable - LOAN_HISTORY';
287
288 PROC FREQ DATA = LIB78400.TRAINING_DS;
289
290 TABLE QUALIFICATION * LOAN_HISTORY /
291 PLOTS = FREQPLOT (TWOWAY = STACKED SCALE = GROUPPCT);
292 RUN;
```

6.3.1.2.2 Screenshot(s)/Output(s)

Bivariate Analysis of the Variables
Categorical Variable - QUALIFICATION VS Categorical Variable - LOAN_HISTORY

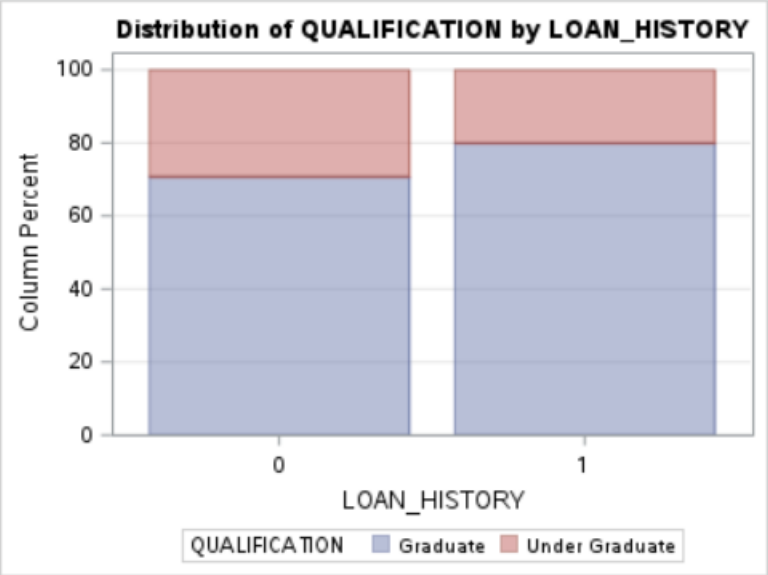
The FREQ Procedure

Frequency
Percent
Row Pct
Col Pct

Table of QUALIFICATION by LOAN_HISTORY

QUALIFICATION	LOAN_HISTORY 0	1	Total
Graduate	63 11.17 14.22 70.79	380 67.38 85.78 80.00	443 78.55
Under Graduate	26 4.61 21.49 29.21	95 16.84 78.51 20.00	121 21.45
Total	89 15.78	475 84.22	564 100.00

Frequency Missing = 50



6.3.1.2.3 Description

The data scientist found that for graduate loan applicants, 67.38% of them had positive past loan records (n = 380) and 11.17% had negative past loan records (n = 63). For undergraduate loan applicants, 16.84% of them had positive past loan records (n = 95) and 4.61% had negative past loan records (n = 26). 50 missing values were found for the bivariate analysis of qualification versus loan history.

6.3.1.3 Bivariate Analysis of the Variables (Categorical – LOAN_HISTORY VS Categorical – LOAN_APPROVAL_STATUS)

6.3.1.3.1 SAS Codes

```
296 TITLE1 'Bivariate Analysis of the Variables';
297 TITLE2 'Categorical Variable - LOAN_HISTORY VS Categorical Variable - LOAN_APPROVAL_STATUS';
298
299 PROC FREQ DATA = LIB78400.TRAINING_DS;
300
301 TABLE LOAN_HISTORY * LOAN_APPROVAL_STATUS /
302 PLOTS = FREQPLOT (TWOWAY = STACKED SCALE = GROUPPCT);
303 RUN;
```

6.3.1.3.2 Screenshot(s)/Output(s)

Bivariate Analysis of the Variables
Categorical Variable - LOAN_HISTORY VS Categorical Variable - LOAN_APPROVAL_STATUS

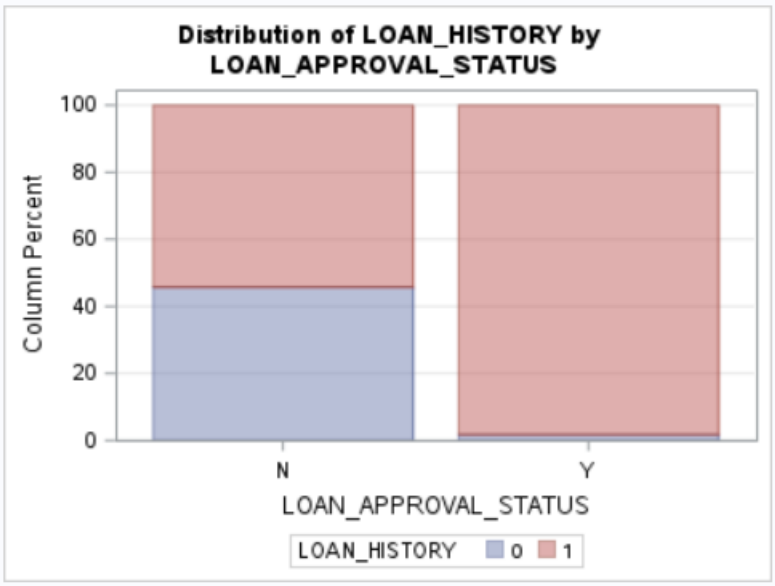
The FREQ Procedure

Frequency
Percent
Row Pct
Col Pct

Table of LOAN_HISTORY by LOAN_APPROVAL_STATUS

LOAN_HISTORY	LOAN_APPROVAL_STATUS N	LOAN_APPROVAL_STATUS Y	Total
0	82 14.54 92.13 45.81	7 1.24 7.87 1.82	89 15.78
1	97 17.20 20.42 54.19	378 67.02 79.58 98.18	475 84.22
Total	179 31.74	385 68.26	564 100.00

Frequency Missing = 50



6.3.1.3.3 Description

The data scientist found that most of the applicants with positive past loan records had their loans approved, representing 67.02% of the population (n = 378). 50 missing values were found for the bivariate analysis of loan history versus loan approval status.

6.3.2 Bivariate Analysis of the Variables (Categorical VS Continuous)

6.3.2.1 Bivariate Analysis of the Variables (Categorical – GENDER VS Continuous – LOAN_AMOUNT)

6.3.2.1.1 SAS Codes

```
308 TITLE1 'Bivariate analysis of the variables: ';
309 TITLE2 'Categorical variable[GENDER] vs Numeric/Continuous variable[LOAN_AMOUNT]';
310
311 PROC MEANS DATA = LIB78400.TRAINING_DS;
312
313 CLASS gender; /* It is a categorial variable */
314 VAR loan_amount; /* it is a Numeric/Continuous variable */
315
316 RUN;
```

6.3.2.1.2 Screenshot(s)/Output(s)

Bivariate analysis of the variables:
Categorical variable[GENDER] vs Numeric/Continuous variable[LOAN_AMOUNT]

The MEANS Procedure

Analysis Variable : LOAN_AMOUNT						
GENDER	N Obs	N	Mean	Std Dev	Minimum	Maximum
Female	112	109	126.6972477	79.2864596	9.0000000	600.0000000
Male	489	470	149.2659574	82.8108508	17.0000000	650.0000000

6.3.2.1.3 Description

The data scientist found that male applicants had a greater average loan amount applied, about 1.2 times more than female applicants. The minimum and maximum loan amounts applied for female applicants were $9,000 and $600,000 respectively, which were less than the minimum and maximum loan amounts for males, which were $17,000 and $650,000 respectively.

6.3.2.2 Bivariate Analysis of the Variables (Categorical – QUALIFICATION VS Continuous – CANDIDATE_INCOME)

6.3.2.2.1 SAS Codes

```
320 TITLE1 'Bivariate analysis of the variables: ';
321 TITLE2 'Categorical variable[QUALIFICATION] vs Numeric/Continuous variable[CANDIDATE_INCOME]';
322
323 PROC MEANS DATA = LIB78400.TRAINING_DS;
324
325 CLASS qualification; /* It is a categorial variable */
326 VAR candidate_income; /* it is a Numeric/Continuous variable */
327
328 RUN;
```

6.3.2.2.2 Screenshot(s)/Output(s)

Bivariate analysis of the variables:
Categorical variable[QUALIFICATION] vs Numeric/Continuous variable[CANDIDATE_INCOME]

The MEANS Procedure

Analysis Variable : CANDIDATE_INCOME						
QUALIFICATION	N Obs	N	Mean	Std Dev	Minimum	Maximum
Graduate	480	480	5857.43	6739.80	150.0000000	81000.00
Under Graduate	134	134	3777.28	2237.08	210.0000000	18165.00

6.3.2.2.3 Description

The data scientist found that graduate applicants had a greater average monthly income, about 1.6 times more than undergraduates. The minimum monthly income for graduate applicants was $150, which was less than the minimum monthly income for graduate applicants, $210. However, the maximum monthly income for graduate applicants was $81,000, which was more than the maximum monthly income for graduate applicants, $18,165.

6.3.2.3 Bivariate Analysis of the Variables (Categorical – LOAN_HISTORY VS Continuous – LOAN_DURATION)

6.3.2.3.1 SAS Codes

```
332 TITLE1 'Bivariate analysis of the variables: ';
333 TITLE2 'Categorical variable[LOAN_HISTORY] vs Numeric/Continuous variable[LOAN_DURATION]';
334
335 PROC MEANS DATA = LIB78400.TRAINING_DS;
336
337 CLASS loan_history; /* It is a categorial variable */
338 VAR loan_duration; /* it is a Numeric/Continuous variable */
339
340 RUN;
```

6.3.2.3.2 Screenshot(s)/Output(s)

Bivariate analysis of the variables:
Categorical variable[LOAN_HISTORY] vs Numeric/Continuous variable[LOAN_DURATION]

The MEANS Procedure

Analysis Variable : LOAN_DURATION						
LOAN_HISTORY	N Obs	N	Mean	Std Dev	Minimum	Maximum
0	89	83	341.9277108	66.7441023	180.0000000	480.0000000
1	475	467	342.1927195	64.2686598	36.0000000	480.0000000

6.3.2.3.3 Description

The data scientist found that there were no significant differences between applicants with negative and positive past loan records in terms of average loan duration. The minimum loan duration for applicants with a negative loan history was 180 months, which was more than the minimum loan duration for applicants with a positive loan history, 36 months. However, the maximum loan duration for applicants with positive and negative loan histories was the same, which was 480 months.

7.0 Analysis of the Variables in the Dataset – LIB78400.TESTING_DS

7.1 Univariate Analysis of the Categorical Variables in the Dataset Using SAS Macro

7.1.1 Introduction

SAS macro is one of the most powerful features of SAS as it can generate code and allow its reuse countless times, thus automating the coding task for data scientists. One can define macro variables and programs at their disposal so that reusable SAS code snippets can be developed with ease. Besides, the structure of the codes can be less complex so that they are more comprehensible and maintainable. For example, performing univariate and bivariate analysis for the training set earlier took many manual coding steps, so SAS macro helps ease this burden by simplifying and automating iterative univariate and bivariate analytical processes. Overall, it contributes towards a more efficient data analysis pipeline.

7.1.2 SAS Codes

```
/* Macro begins here */
OPTIONS MCOMPILENOTE = ALL;

%MACRO UVA_CATE_VARI(ptitle, pdataset, pcate_vari);

TITLE &ptitle;

PROC FREQ DATA = &pdataset;

TABLE &pcate_vari;

RUN;

%MEND UVA_CATE_VARI;
/* Macro ends here */
```

```
/* Call the SAS Macro - UVA_CATE_VARI */
%UVA_CATE_VARI('Univariate Analysis of the Categorical variable - MARITAL_STATUS', LIB78400.TESTING_DS, MARITAL_STATUS);
%UVA_CATE_VARI('Univariate Analysis of the Categorical variable - GENDER', LIB78400.TESTING_DS, GENDER);
%UVA_CATE_VARI('Univariate Analysis of the Categorical variable - LOAN_HISTORY', LIB78400.TESTING_DS, LOAN_HISTORY);
%UVA_CATE_VARI('Univariate Analysis of the Categorical variable - FAMILY_MEMBERS', LIB78400.TESTING_DS, FAMILY_MEMBERS);
%UVA_CATE_VARI('Univariate Analysis of the Categorical variable - QUALIFICATION', LIB78400.TESTING_DS, QUALIFICATION);
%UVA_CATE_VARI('Univariate Analysis of the Categorical variable - EMPLOYMENT', LIB78400.TESTING_DS, EMPLOYMENT);
%UVA_CATE_VARI('Univariate Analysis of the Categorical variable - LOAN_LOCATION', LIB78400.TESTING_DS, LOAN_LOCATION);
```

7.1.3 Screenshot(s)/Output(s)

Univariate Analysis of the Categorical variable - MARITAL_STATUS

The FREQ Procedure

MARITAL_STATUS	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Married	233	63.49	233	63.49
Not Married	134	36.51	367	100.00

Univariate Analysis of the Categorical variable - GENDER

The FREQ Procedure

GENDER	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Female	70	19.66	70	19.66
Male	286	80.34	356	100.00
Frequency Missing = 11				

11 missing values found for GENDER

Univariate Analysis of the Categorical variable - LOAN_HISTORY

The FREQ Procedure

LOAN_HISTORY	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	59	17.46	59	17.46
1	279	82.54	338	100.00
Frequency Missing = 29				

29 missing values found for LOAN_HISTORY

10 missing values found for FAMILY_MEMBERS

Univariate Analysis of the Categorical variable - FAMILY_MEMBERS

The FREQ Procedure

FAMILY_MEMBERS	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	200	56.02	200	56.02
1	58	16.25	258	72.27
2	59	16.53	317	88.80
3+	40	11.20	357	100.00
Frequency Missing = 10				

Univariate Analysis of the Categorical variable - QUALIFICATION

The FREQ Procedure

QUALIFICATION	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Graduate	283	77.11	283	77.11
Under Graduate	84	22.89	367	100.00

23 missing values found for EMPLOYMENT

Univariate Analysis of the Categorical variable - EMPLOYMENT

The FREQ Procedure

EMPLOYMENT	Frequency	Percent	Cumulative Frequency	Cumulative Percent
No	307	89.24	307	89.24
Yes	37	10.76	344	100.00
Frequency Missing = 23				

Univariate Analysis of the Categorical variable - LOAN_LOCATION

The FREQ Procedure

LOAN_LOCATION	Frequency	Percent	Cumulative Frequency	Cumulative Percent
City	140	38.15	140	38.15
Town	116	31.61	256	69.75
Village	111	30.25	367	100.00

7.1.4 Description

In the testing set, 63.49% of loan applicants were married (n = 233) and 36.51% of them were not married (n = 134), with no values missing in this MARITAL_STATUS variable.

80.34 % of loan applicants were male (n = 286) and 19.66% of them were female (n = 70), with 11 values missing in this GENDER variable.

82.54 % of loan applicants have positive past loan records (n = 279) and 17.46 % of them have negative past loan records (n = 59), with 29 values missing in this LOAN_HISTORY variable.

Most applicants had no family members (n = 200), slightly followed by two (n = 59), one (n = 58) and three or more family members (n = 40), with 10 values missing in this FAMILY_MEMBERS variable.

77.11% of loan applicants were graduates (n = 283) and 22.89% were undergraduates (n = 84), with no values missing in this QUALIFICATION variable.

89.24% of loan applicants were non-employed (n = 307) and 10.76% were employed (n = 37), with 23 values missing in this EMPLOYMENT variable.

Most applicants are in the city (n = 140), slightly followed by town (n = 116) and village (n = 111), with no values missing in this LOAN_LOCATION variable.

7.2 Univariate Analysis of the Continuous Variables in the Dataset Using SAS Macro

7.2.1 SAS Codes

```
369 /* Macro begins here */
370 OPTIONS MCOMPILENOTE = ALL;
371
372 %MACRO UVA_CONTI_VARI(ptitle, pdataset, pconti_vari);
373
374 TITLE &ptitle;
375
376 PROC MEANS DATA = &pdataset N NMISS MIN MAX MEAN MEDIAN STD;
377
378 VAR &pconti_vari;
379
380 RUN;
381
382 %MEND UVA_CONTI_VARI;
383 /* MACRO ends here */
```

```
385 /* Call the SAS MACRO - UVA_CONTI_VARI */
386 %UVA_CONTI_VARI('UVA of the continuous variable - CANDIDATE_INCOME', LIB78400.TESTING_DS, candidate_income);
387 %UVA_CONTI_VARI('UVA of the continuous variable - GUARANTEE_INCOME', LIB78400.TESTING_DS, guarantee_income);
388 %UVA_CONTI_VARI('UVA of the continuous variable - LOAN_AMOUNT', LIB78400.TESTING_DS, loan_amount);
389 %UVA_CONTI_VARI('UVA of the continuous variable - LOAN_DURATION', LIB78400.TESTING_DS, loan_duration);
```

7.2.2 Screenshot(s)/Output(s)

UVA of the continuous variable - CANDIDATE_INCOME

The MEANS Procedure

Analysis Variable : CANDIDATE_INCOME						
N	N Miss	Minimum	Maximum	Mean	Median	Std Dev
367	0	0	72529.00	4805.60	3786.00	4910.69

UVA of the continuous variable - GUARANTEE_INCOME

The MEANS Procedure

Analysis Variable : GUARANTEE_INCOME						
N	N Miss	Minimum	Maximum	Mean	Median	Std Dev
367	0	0	24000.00	1569.58	1025.00	2334.23

UVA of the continuous variable - LOAN_AMOUNT

The MEANS Procedure

Analysis Variable : LOAN_AMOUNT						
N	N Miss	Minimum	Maximum	Mean	Median	Std Dev
362	5	28.0000000	550.0000000	136.1325967	125.0000000	61.3666524

UVA of the continuous variable - LOAN_DURATION

The MEANS Procedure

Analysis Variable : LOAN_DURATION						
N	N Miss	Minimum	Maximum	Mean	Median	Std Dev
361	6	6.0000000	480.0000000	342.5373961	360.0000000	65.1566434

7.2.3 Description

In the testing set, the typical and average monthly income of all applicants were $3,786 and $4805.60 respectively, with no values missing in this CANDIDATE_INCOME variable.

The typical and average monthly joint income of all applicants were $1,025 and $1569.58 respectively, with no values missing in this GUARANTEE_INCOME variable.

The typical and average loan amounts of all applicants were $125,000 and $136,133 respectively, with five values missing in this LOAN_AMOUNT variable.

The typical and average loan durations of all applicants were 360 and 343 months respectively, with six values missing in this LOAN_DURATION variable.

7.3 Bivariate Analysis of the Variables in the Dataset

7.3.1 Bivariate Analysis of the Variables (Categorical VS Categorical) Using SAS Macro

7.3.1.1 SAS Codes

```
/* Macro begins here */
OPTIONS MCOMPILENOTE = ALL;

%MACRO BVA_CATE_CATE(ptitle1, ptitle2, pdataset, pcate_vari1, pcate_vari2);

TITLE1 &ptitle1;
TITLE2 &ptitle2;

PROC FREQ DATA = &pdataset;

TABLE &pcate_vari1 * &pcate_vari2/
PLOTS = FREQPLOT( TWOWAY = STACKED SCALE = GROUPPCT );

RUN;
%MEND BVA_CATE_CATE;
/* MACRO ends here */
```

```
/* Call the SAS MACRO - BVA_CATE_CATE */
%BVA_CATE_CATE('Bivariate Analysis of the Variables: ',
'MARITAL_STATUS VS EMPLOYMENT',
LIB78400.TESTING_DS,
MARITAL_STATUS, EMPLOYMENT);

%BVA_CATE_CATE('Bivariate Analysis of the Variables: ',
'GENDER VS EMPLOYMENT',
LIB78400.TESTING_DS,
GENDER, EMPLOYMENT);

%BVA_CATE_CATE('Bivariate Analysis of the Variables: ',
'QUALIFICATION VS EMPLOYMENT',
LIB78400.TESTING_DS,
QUALIFICATION, EMPLOYMENT);
```

7.3.1.2 Screenshot(s)/Output(s)

Bivariate Analysis of the Variables: MARITAL_STATUS VS EMPLOYMENT

The FREQ Procedure

Frequency
Percent
Row Pct
Col Pct

Table of MARITAL_STATUS by EMPLOYMENT

MARITAL_STATUS	EMPLOYMENT No	EMPLOYMENT Yes	Total
Married	192 55.81 88.48 62.54	25 7.27 11.52 67.57	217 63.08
Not Married	115 33.43 90.55 37.46	12 3.49 9.45 32.43	127 36.92
Total	307 89.24	37 10.76	344 100.00

Frequency Missing = 23

Bivariate Analysis of the Variables: GENDER VS EMPLOYMENT

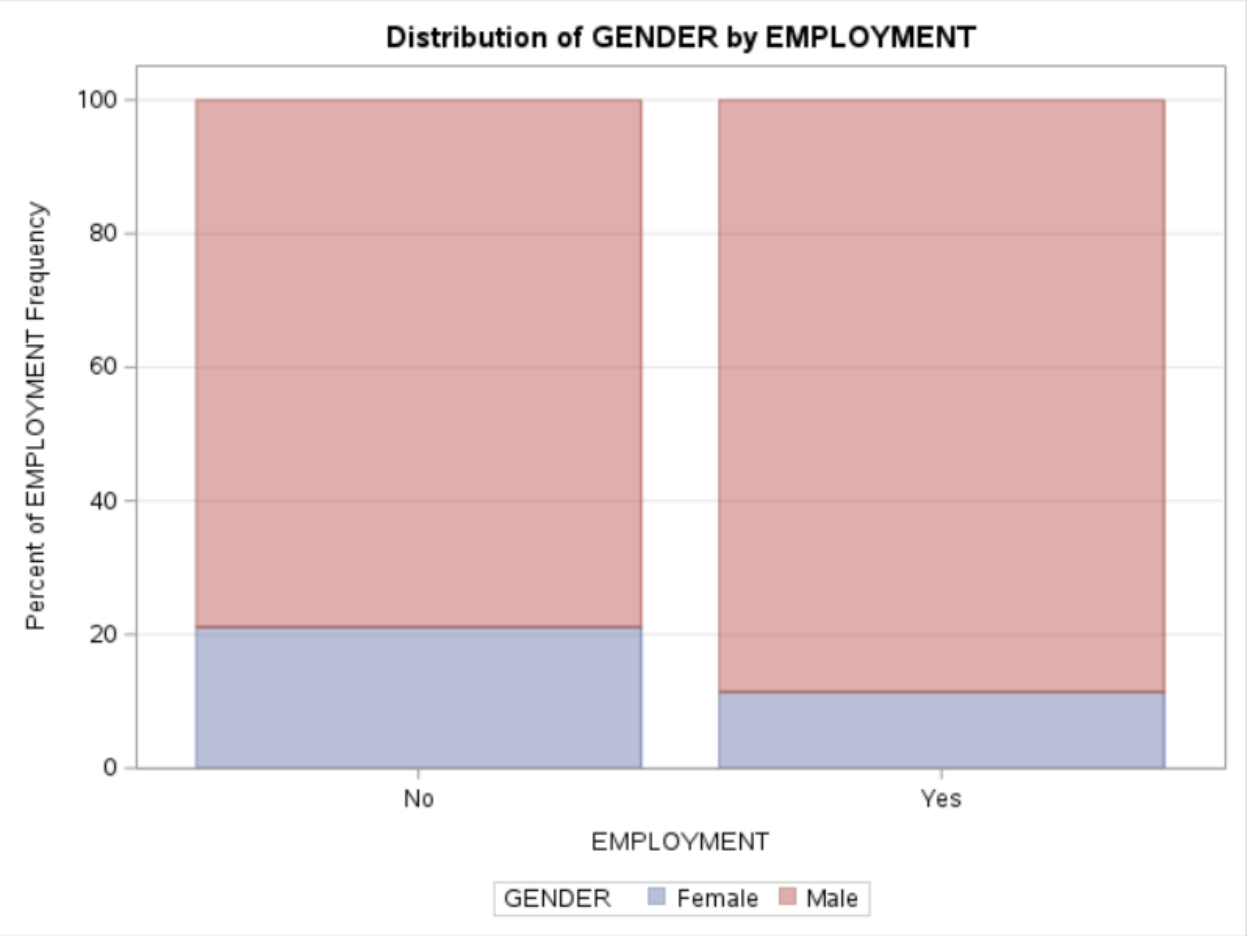
The FREQ Procedure

Frequency
Percent
Row Pct
Col Pct

Table of GENDER by EMPLOYMENT

GENDER	EMPLOYMENT No	EMPLOYMENT Yes	Total
Female	63 18.92 94.03 21.14	4 1.20 5.97 11.43	67 20.12
Male	235 70.57 88.35 78.86	31 9.31 11.65 88.57	266 79.88
Total	298 89.49	35 10.51	333 100.00

Frequency Missing = 34



Bivariate Analysis of the Variables: QUALIFICATION VS EMPLOYMENT

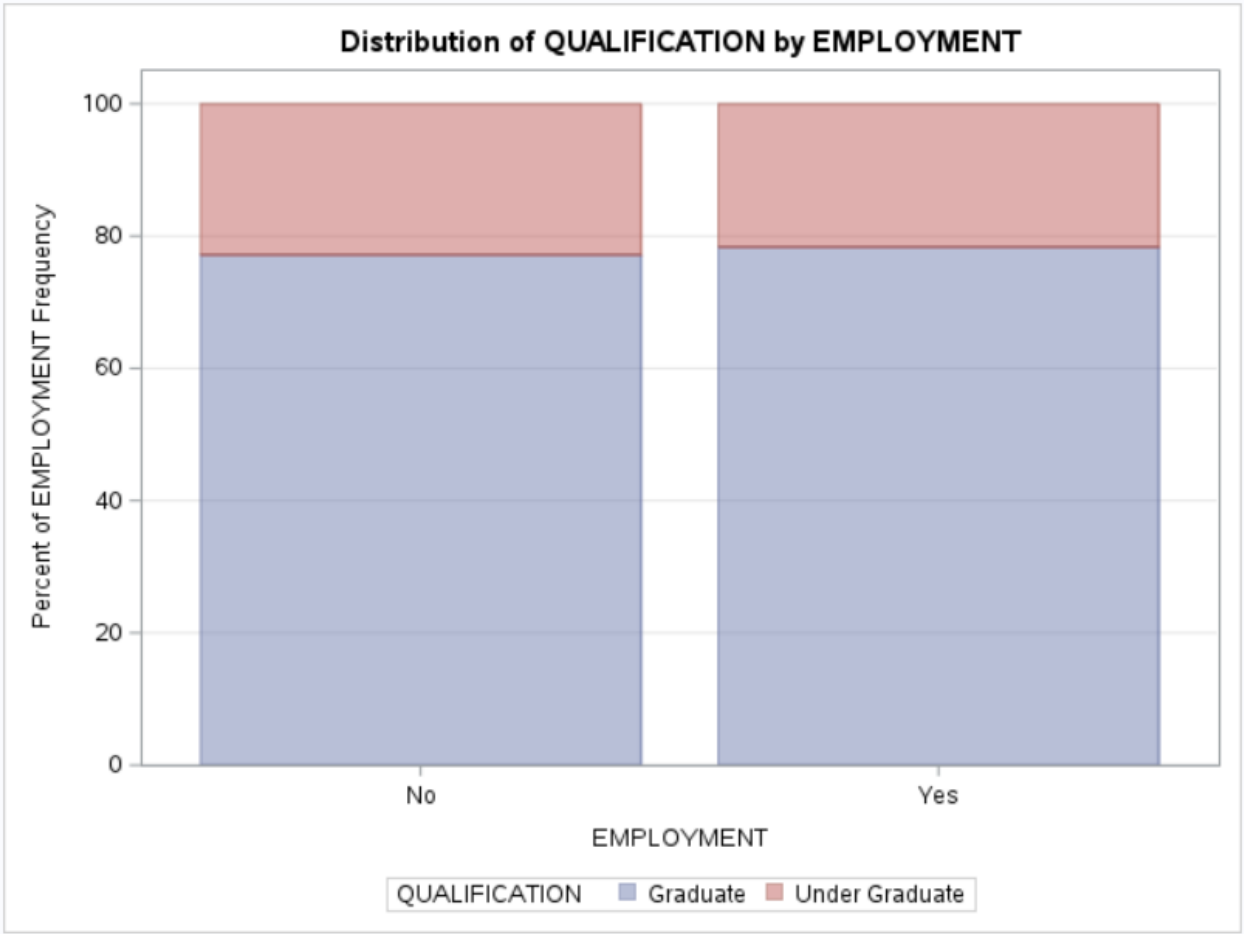
The FREQ Procedure

Frequency
Percent
Row Pct
Col Pct

Table of QUALIFICATION by EMPLOYMENT

QUALIFICATION	EMPLOYMENT No	EMPLOYMENT Yes	Total
Graduate	237 68.90 89.10 77.20	29 8.43 10.90 78.38	266 77.33
Under Graduate	70 20.35 89.74 22.80	8 2.33 10.26 21.62	78 22.67
Total	307 89.24	37 10.76	344 100.00

Frequency Missing = 23



7.3.1.3 Description

For applicants with no employment, 55.81% of them were married (n = 192) and 33.43% were not married (n = 115). For applicants with employment, 7.27% of them were married (n = 25) and 3.49% were not married (n = 12). 23 missing values were found for the bivariate analysis of marital status versus employment.

For applicants with no employment, 18.92% of them were female (n = 63) and 70.57% were male (n = 235). For applicants with employment, 7.27% of them were female (n = 4) and 9.31% were male (n = 31). 34 missing values were found for the bivariate analysis of gender versus employment.

For applicants with no employment, 68.9% of them were graduates (n = 237) and 20.35 % were undergraduates (n = 70). For applicants with employment, 8.43% of them were graduates (n = 29) and 2.33% were undergraduates (n = 8). 23 missing values were found for the bivariate analysis of qualification versus employment.

7.3.2 Bivariate Analysis of the Variables (Categorical VS Continuous) Using SAS Macro

7.3.2.1 SAS Codes

```
426 /* Categorical VS Continuous */
427 /* Macro begins here */
428 OPTIONS MCOMPILENOTE = ALL;
429 %MACRO BVA_CATE_CONTI(ptitle1, ptitle2, pdataset, pcate_vari, pconti_vari);
430 TITLE1 &ptitle1;
431 TITLE2 &ptitle2;
432
433 PROC MEANS DATA = &pdataset;
434
435 CLASS &pcate_vari;  /* It is a categorical variable */
436 VAR &pconti_vari; /* It is a numeric/continuous variable */
437
438 RUN;
439 %MEND BVA_CATE_CONTI;
440 /* Macro ends here */
```

```
442 /* Call the SAS MACRO - BVA_CATE_CONTI */
443
444 %BVA_CATE_CONTI('Bivariate Analysis of the Variables (Categorical VS Continuous -',
445 'MARITAL_STATUS VS LOAN_AMOUNT)',
446 LIB78400.TESTING_DS, MARITAL_STATUS, LOAN_AMOUNT);
447
448 %BVA_CATE_CONTI('Bivariate Analysis of the Variables (Categorical vs Continuous -',
449 'EMPLOYMENT VS CANDIDATE_INCOME)',
450 LIB78400.TESTING_DS, EMPLOYMENT, CANDIDATE_INCOME);
451
452 %BVA_CATE_CONTI('Bivariate Analysis of the Variables (Categorical vs Continuous -',
453 'FAMILY_MEMBERS VS LOAN_DURATION)',
454 LIB78400.TESTING_DS, FAMILY_MEMBERS, LOAN_DURATION);
```

7.3.2.2 Screenshot(s)/Output(s)

Bivariate Analysis of the Variables (Categorical VS Continuous - MARITAL_STATUS VS LOAN_AMOUNT)

The MEANS Procedure

Analysis Variable : LOAN_AMOUNT						
MARITAL_STATUS	N Obs	N	Mean	Std Dev	Minimum	Maximum
Married	233	228	144.6754386	67.7425153	28.0000000	550.0000000
Not Married	134	134	121.5970149	45.2903946	28.0000000	300.0000000

Bivariate Analysis of the Variables (Categorical vs Continuous - EMPLOYMENT VS CANDIDATE_INCOME)

The MEANS Procedure

Analysis Variable : CANDIDATE_INCOME						
EMPLOYMENT	N Obs	N	Mean	Std Dev	Minimum	Maximum
No	307	307	4635.16	4922.29	0	72529.00
Yes	37	37	5874.86	5293.83	570.0000000	32000.00

Bivariate Analysis of the Variables (Categorical vs Continuous - FAMILY_MEMBERS VS LOAN_DURATION)

The MEANS Procedure

Analysis Variable : LOAN_DURATION						
FAMILY_MEMBERS	N Obs	N	Mean	Std Dev	Minimum	Maximum
0	200	197	345.0558376	60.9019112	6.0000000	480.0000000
1	58	56	346.5000000	60.7184261	60.0000000	480.0000000
2	59	59	340.8813559	72.4852148	12.0000000	480.0000000
3+	40	39	330.7692308	75.2324334	120.0000000	480.0000000

7.3.2.3 Description

The data scientist found that the average loan amount for married applicants was greater than for non-married applicants. The minimum loan amount for married and non-married applicants was the same, which was $28,000. However, the maximum loan amount for married applicants, $550,000 was greater than non-married applicants, $300,000.

The average monthly income for non-employed applicants was less than employed applicants. The minimum monthly income for non-employed applicants was zero while for employed applicants was $570. However, the maximum loan amount for non-employed applicants, $72,529 was greater than employed applicants, $32,000.

Applicants with one family member had the greatest average loan duration, 346 and half months, slightly followed by zero, two and three or more family members. However, the maximum loan amount for applicants was 480 months regardless of the number of family members.

8.0 Imputing Missing Values Found in the Categorical Variables of LIB78400.TRAINING_DS

8.1 Imputing Missing Values Found in the Categorical Variable – MARITAL_STATUS

8.1.1 Step 1: List Out the Loan Applicants Who Did Not Detail Their Marital Status During the Loan Application Submission

8.1.1.1 SAS Codes

```
458 /* STEP 1: Find the details of loan applicants who submitted
459 their loan application without marital status */
460 TITLE1 'Find the details of loan applicants who submitted';
461 TITLE2 'their loan application without marital status';
462 FOOTNOTE '----------END----------';
463
464 PROC SQL;
465
466 SELECT *
467 FROM LIB78400.TRAINING_DS e
468 WHERE ( e.MARITAL_STATUS eq '' OR e.marital_status IS MISSING );
469
470 QUIT;
```

8.1.1.2 Screenshot(s)/Output(s)

Find the details of loan applicants who submitted their loan application without marital status

SME_LOAN_ID_NO	GENDER	MARITAL_STATUS	FAMILY_MEMBERS	QUALIFICATION	EMPLOYMENT	CANDIDATE_INCOME	GUARANTEE_INCOME	LOAN_AMOUNT	LOAN_DURATION	LOAN_HISTORY	LOAN_LOCATION	LOAN_APPROVAL_STATUS
LP001357	Male			Graduate	No	3816	754	160	360	1	City	Y
LP001760	Male			Graduate	No	4758	0	158	480	1	Town	Y
LP002393	Female			Graduate	No	10047	0	.	240	1	Town	Y

----------END----------

8.1.1.3 Description

The data scientist would like to discover which loan applicants did not specify their marital status during the application submission stage. There were two male and one female graduate and unemployed applicants with positive loan records who did not specify their marital status. Their monthly income ranges between $3,816 and $10,047 and loan duration ranges between 240 and 480 months.

8.1.2 Step 2: Count the Loan Applicants Who Did Not Detail Their Marital Status During the Loan Application Submission

8.1.2.1 SAS Codes

```
472 /* STEP 2: Count the number of loan applicants who submitted
473 their loan application without marital status */
474 TITLE1 'Count the number of loan applicants who submitted';
475 TITLE2 'their loan application without marital status';
476 FOOTNOTE '----------END----------';
477
478 PROC SQL;
479
480 SELECT COUNT(*) Label = 'Number of Applicants'
481 FROM LIB78400.TRAINING_DS e
482 WHERE ( e.MARITAL_STATUS eq '' OR e.marital_status IS MISSING );
483
484 QUIT;
```

8.1.2.2 Screenshot(s)/Output(s)

Count the number of loan applicants who submitted their loan application without marital status

Number of Applicants
3

----------END----------

8.1.2.3 Description

The data scientist would like to calculate the number of loan applicants who did not specify their marital status during the application submission stage. As a result, three applicants did not specify their marital status.

8.1.3 Step 3: Find the Statistics of Married and Non-married Loan Applicants

8.1.3.1 SAS Codes

```
/* STEP 3: Find the statistics of married & non-married applicants */
TITLE1 'Find the statistics of married & non-married applicants';

PROC SQL;

SELECT e.MARITAL_STATUS AS MARITAL_STATUS,
       COUNT(*) AS COUNTS
FROM LIB78400.TRAINING_DS e
WHERE ( e.MARITAL_STATUS ne '' OR e.MARITAL_STATUS IS NOT MISSING )
GROUP BY e.MARITAL_STATUS;

QUIT;
```

8.1.3.2 Screenshot(s)/Output(s)

Find the statistics of married & non-married applicants

MARITAL_STATUS	COUNTS
Married	398
Not Married	213

8.1.3.3 Description

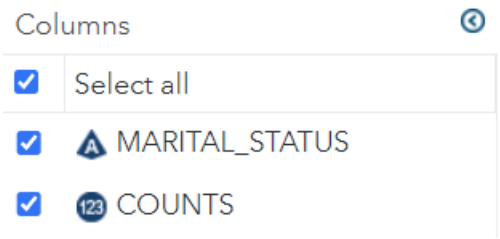
The data scientist would like to discover the statistics of married and non-married applicants. As a result, there were 398 married applicants and 213 non-married applicants.

8.1.4 Step 4: Save the Statistics of Married and Non-married Loan Applicants in a Dataset

8.1.4.1 SAS Codes

```
499 /* STEP 4: Save the statistics in a dataset */
500
501 PROC SQL;
502
503 CREATE TABLE LIB78400.TRAINING_STAT_DS AS
504 SELECT e.MARITAL_STATUS AS MARITAL_STATUS,
505        COUNT(*) AS COUNTS
506 FROM LIB78400.TRAINING_DS e
507 WHERE ( e.MARITAL_STATUS ne '' OR e.MARITAL_STATUS IS NOT MISSING )
508 GROUP BY e.MARITAL_STATUS;
509
510 QUIT;
```

8.1.4.2 Screenshot(s)/Output(s)



Total rows: 2 Total columns: 2

	MARITAL_STA...	COUNTS
1	Married	398
2	Not Married	213

8.1.4.3 Description

The data scientist would like to save the statistics of married and non-married applicants into a new dataset that would be stored in the LIB78400 library. As a result, a new dataset called LIB78400.TRAINING_STAT_DS was created.

8.1.5 Step 4.1: Make a Backup Copy of the Dataset Created

8.1.5.1 SAS Codes

```
512 /* STEP 4.1: Make a backup copy of the dataset */
513
514 PROC SQL;
515
516 CREATE TABLE LIB78400.TRAINING_BK_DS AS
517 SELECT *
518 FROM LIB78400.TRAINING_DS;
519
520 QUIT;
```

8.1.5.2 Screenshot(s)/Output(s)

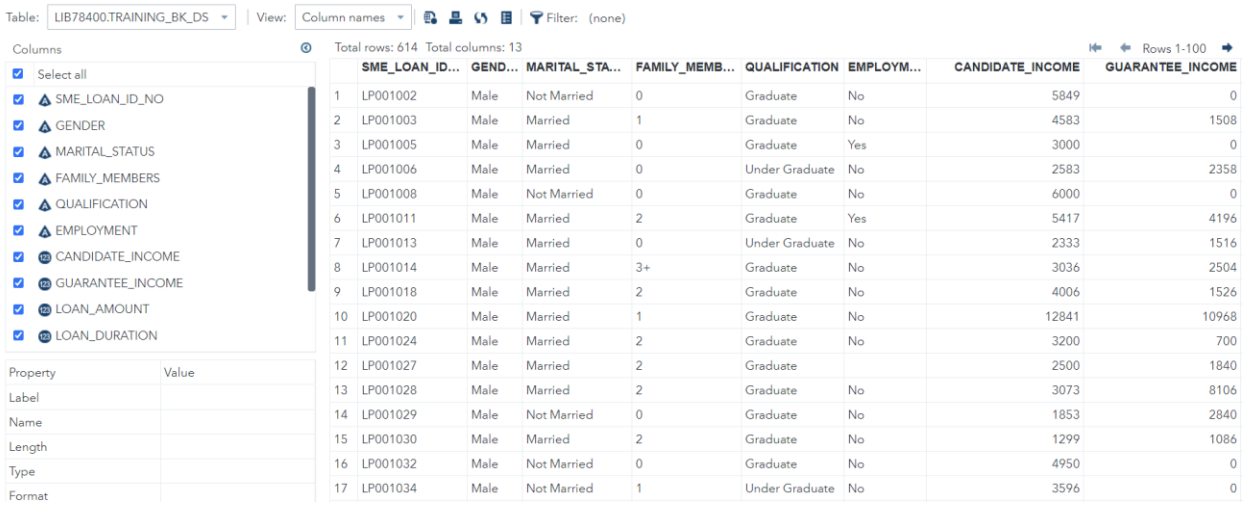


Table: LIB78400.TRAINING_BK_DS | View: Column names | Filter: (none)

Columns: Select all; SME_LOAN_ID_NO; GENDER; MARITAL_STATUS; FAMILY_MEMBERS; QUALIFICATION; EMPLOYMENT; CANDIDATE_INCOME; GUARANTEE_INCOME; LOAN_AMOUNT; LOAN_DURATION

Property	Value
Label	
Name	
Length	
Type	
Format	

Total rows: 614 Total columns: 13 — Rows 1-100

	SME_LOAN_ID...	GEND...	MARITAL_STA...	FAMILY_MEMB...	QUALIFICATION	EMPLOYM...	CANDIDATE_INCOME	GUARANTEE_INCOME
1	LP001002	Male	Not Married	0	Graduate	No	5849	0
2	LP001003	Male	Married	1	Graduate	No	4583	1508
3	LP001005	Male	Married	0	Graduate	Yes	3000	0
4	LP001006	Male	Married	0	Under Graduate	No	2583	2358
5	LP001008	Male	Not Married	0	Graduate	No	6000	0
6	LP001011	Male	Married	2	Graduate	Yes	5417	4196
7	LP001013	Male	Married	0	Under Graduate	No	2333	1516
8	LP001014	Male	Married	3+	Graduate	No	3036	2504
9	LP001018	Male	Married	2	Graduate	No	4006	1526
10	LP001020	Male	Married	1	Graduate	No	12841	10968
11	LP001024	Male	Married	2	Graduate	No	3200	700
12	LP001027	Male	Married	2	Graduate		2500	1840
13	LP001028	Male	Married	2	Graduate	No	3073	8106
14	LP001029	Male	Not Married	0	Graduate	No	1853	2840
15	LP001030	Male	Married	2	Graduate	No	1299	1086
16	LP001032	Male	Not Married	0	Graduate	No	4950	0
17	LP001034	Male	Not Married	1	Under Graduate	No	3596	0

8.1.5.3 Description

The data scientist would like to make a backup copy of the training set so that he can access it whenever the original training set is corrupted during analysis. As a result, a backup copy called LIB78400.TRAINING_BK_DS was created.

8.1.6 Step 5: Impute Missing Values in the Categorical Variable – MARITAL_STATUS

8.1.6.1 SAS Codes

```
/* STEP 5: Impute the missing values found in the categorical variable - MARITAL_STATUS */

PROC SQL;

UPDATE LIB78400.TRAINING_DS
SET MARITAL_STATUS = ( SELECT to.MARITAL_STATUS AS MARITAL_STATUS
                       FROM LIB78400.TRAINING_STAT_DS to
                       WHERE to.COUNTS eq ( SELECT MAX(ti.COUNTS) AS HIGHEST_COUNT
                                            FROM LIB78400.TRAINING_STAT_DS ti ) )
                                            /* Above is a sub-program to find the highest count */
WHERE ( MARITAL_STATUS eq '' OR marital_status IS MISSING );

QUIT;
```

8.1.6.2 Screenshot(s)/Output(s)

```
NOTE: 3 rows were updated in LIB78400.TRAINING_DS.
```

8.1.6.3 Description

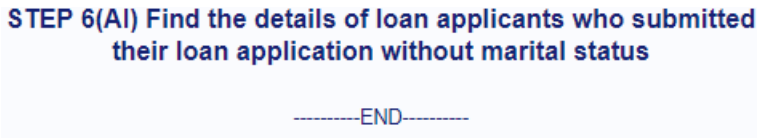
The data scientist would like to impute the three missing values found in MARITAL_STATUS with its mode, married. As a result, the three rows with missing values for MARITAL_STATUS were updated with married status in LIB78400.TRAINING_DS.

8.1.7 Step 6: List Out the Loan Applicants Who Did Not Detail Their Marital Status During the Loan Application Submission (After Imputation)

8.1.7.1 SAS Codes

```
536 /* STEP 6: (AI) Find the details of loan applicants who submitted their loan application without marital status */
537 TITLE1 'STEP 6(AI) Find the details of loan applicants who submitted';
538 TITLE2 'their loan application without marital status';
539 FOOTNOTE '----------END----------';
540
541 PROC SQL;
542
543 SELECT *
544 FROM LIB78400.TRAINING_DS e
545 WHERE ( e.MARITAL_STATUS eq '' OR e.marital_status IS MISSING );
546
547 QUIT;
```

8.1.7.2 Screenshot(s)/Output(s)



8.1.7.3 Description

After imputation, the data scientist would like to double-check if the details of loan applicants who did not specify their marital status were still there. As a result, the details could not be found as the missing values for MARITAL_STATUS were imputed with ‘married’.

8.2 Imputing Missing Values Found in the Categorical Variable – GENDER

8.2.1 Step 1: List Out the Loan Applicants Who Did Not Detail Their Gender During the Loan Application Submission

8.2.1.1 SAS Codes

```
550 /* STEP 1: Find the details of loan applicants who submitted
551 their loan application without specifying their gender */
552 TITLE1 'Find the details of loan applicants who submitted';
553 TITLE2 'their loan application without specifying their gender';
554 FOOTNOTE '----------END----------';
555
556 PROC SQL;
557
558 SELECT *
559 FROM LIB78400.TRAINING_DS e
560 WHERE ( e.GENDER eq '' OR e.GENDER IS MISSING );
561
562 QUIT;
```

8.2.1.2 Screenshot(s)/Output(s)

Find the details of loan applicants who submitted
their loan application without specifying their gender

SME_LOAN_ID_NO	GENDER	MARITAL_STATUS	FAMILY_MEMBERS	QUALIFICATION	EMPLOYMENT	CANDIDATE_INCOME	GUARANTEE_INCOME	LOAN_AMOUNT	LOAN_DURATION	LOAN_HISTORY	LOAN_LOCATION	LOAN_APPROVAL_STATUS
LP001050		Married	2	Under Graduate	No	3365	1917	112	360	0	Village	N
LP001448		Married	3+	Graduate	No	23803	0	370	360	1	Village	Y
LP001585		Married	3+	Graduate	No	51763	0	700	300	1	City	Y
LP001644		Married	0	Graduate	Yes	674	5296	168	360	1	Village	Y
LP002024		Married	0	Graduate	No	2473	1843	159	360	1	Village	N
LP002103		Married	1	Graduate	Yes	9833	1833	182	180	1	City	Y
LP002478		Married	0	Graduate	Yes	2083	4083	160	360	.	Town	Y
LP002501		Married	0	Graduate	No	16692	0	110	360	1	Town	Y
LP002530		Married	2	Graduate	No	2873	1872	132	360	0	Town	N
LP002625		Not Married	0	Graduate	No	3583	0	96	360	1	City	N
LP002872		Married	0	Graduate	No	3087	2210	136	360	0	Town	N
LP002925		Not Married	0	Graduate	No	4750	0	94	360	1	Town	Y
LP002933		Not Married	3+	Graduate	Yes	9357	0	292	360	1	Town	Y

----------END----------

8.2.1.3 Description

The data scientist would like to discover which loan applicants did not specify their gender during the application submission stage. There were 10 married and three non-married applicants who were mostly graduates, non-employed and had 360 months of loan repayment did not specify their gender.

8.2.2 Step 2: Count the Loan Applicants Who Did Not Detail Their Gender During the Loan Application Submission

8.2.2.1 SAS Codes

```
564 /* STEP 2: Count the number of loan applicants who submitted
565 their loan application without specifying their gender */
566 TITLE1 'Count the number of loan applicants who submitted';
567 TITLE2 'their loan application without without specifying their gender';
568 FOOTNOTE '----------END----------';
569
570 PROC SQL;
571
572 SELECT COUNT(*) Label = 'Number of Applicants'
573 FROM LIB78400.TRAINING_DS e
574 WHERE ( e.GENDER eq '' OR e.GENDER IS MISSING );
575
576 QUIT;
```

8.2.2.2 Screenshot(s)/Output(s)

Count the number of loan applicants who submitted
their loan application without without specifying their gender

Number of Applicants
13

----------END----------

8.2.2.3 Description

The data scientist would like to calculate the number of loan applicants who did not specify their gender during the application submission stage. As a result, 13 applicants did not specify their gender.

8.2.3 Step 3: Find the Statistics of Male and Female Applicants

8.2.3.1 SAS Codes

```
578 /* STEP 3: Find the statistics of male and female applicants */
579 TITLE1 'Find the statistics of male & female applicants';
580
581 PROC SQL;
582
583 SELECT e.GENDER AS GENDER,
584        COUNT(*) AS COUNTS
585 FROM LIB78400.TRAINING_DS e
586 WHERE ( e.GENDER ne '' OR e.GENDER IS NOT MISSING )
587 GROUP BY e.GENDER;
```

8.2.3.2 Screenshot(s)/Output(s)

Find the statistics of male & female applicants

GENDER	COUNTS
Female	112
Male	489

8.2.3.3 Description

The data scientist would like to discover the statistics of male and female applicants. As a result, there were 112 female applicants and 489 male applicants.

8.2.4 Step 4: Save the Statistics of Male and Female Applicants in a Dataset

8.2.4.1 SAS Codes

```
591 /* STEP 4: Save the statistics in a dataset */
592
593 PROC SQL;
594
595 CREATE TABLE LIB78400.TRAINING_GENDER_STAT_DS AS
596 SELECT e.GENDER AS GENDER,
597        COUNT(*) AS COUNTS
598 FROM LIB78400.TRAINING_DS e
599 WHERE ( e.GENDER ne '' OR e.GENDER IS NOT MISSING )
600 GROUP BY e.GENDER;
601
602 QUIT;
```

8.2.4.2 Screenshot(s)/Output(s)

Columns

- Select all
- GENDER
- COUNTS

Total rows: 2 Total columns: 2 Rows 1-2

	GENDER	COUNTS
1	Female	112
2	Male	489

8.2.4.3 Description

The data scientist would like to save the statistics of female and male applicants into a new dataset that would be stored in the LIB78400 library. As a result, a new dataset called LIB78400.TRAINING_GENDER_STAT_DS was created.

8.2.5 Step 4.1: Make a Backup Copy of the Dataset Created

8.2.5.1 SAS Codes

```
604 /* STEP 4.1: Make a backup copy of the dataset */
605
606 PROC SQL;
607
608 CREATE TABLE LIB78400.TRAINING_BK_DS AS
609 SELECT *
610 FROM LIB78400.TRAINING_DS;
611
612 QUIT;
```

8.2.5.2 Screenshot(s)/Output(s)

Table: LIB78400.TRAINING_BK_DS | View: Column names | Filter: (none)

Columns

- Select all
- SME_LOAN_ID_NO
- GENDER
- MARITAL_STATUS
- FAMILY_MEMBERS
- QUALIFICATION
- EMPLOYMENT
- CANDIDATE_INCOME
- GUARANTEE_INCOME
- LOAN_AMOUNT
- LOAN_DURATION

Property	Value
Label	
Name	
Length	
Type	
Format	

Total rows: 614 Total columns: 13 Rows 1-100

	SME_LOAN_ID...	GEND...	MARITAL_STA...	FAMILY_MEMB...	QUALIFICATION	EMPLOYM...	CANDIDATE_INCOME	GUARANTEE_INCOME
1	LP001002	Male	Not Married	0	Graduate	No	5849	0
2	LP001003	Male	Married	1	Graduate	No	4583	1508
3	LP001005	Male	Married	0	Graduate	Yes	3000	0
4	LP001006	Male	Married	0	Under Graduate	No	2583	2358
5	LP001008	Male	Not Married	0	Graduate	No	6000	0
6	LP001011	Male	Married	2	Graduate	Yes	5417	4196
7	LP001013	Male	Married	0	Under Graduate	No	2333	1516
8	LP001014	Male	Married	3+	Graduate	No	3036	2504
9	LP001018	Male	Married	2	Graduate	No	4006	1526
10	LP001020	Male	Married	1	Graduate	No	12841	10968
11	LP001024	Male	Married	2	Graduate	No	3200	700
12	LP001027	Male	Married	2	Graduate		2500	1840
13	LP001028	Male	Married	2	Graduate	No	3073	8106
14	LP001029	Male	Not Married	0	Graduate	No	1853	2840
15	LP001030	Male	Married	2	Graduate	No	1299	1086
16	LP001032	Male	Not Married	0	Graduate	No	4950	0
17	LP001034	Male	Not Married	1	Under Graduate	No	3596	0

8.2.5.3 Description

The data scientist would like to make a backup copy of the training set so that he can access it whenever the original training set was corrupted during analysis. As a result, a backup copy called LIB78400.TRAINING_BK_DS was created.

8.2.6 Step 5: Impute Missing Values in the Categorical Variable – GENDER

8.2.6.1 SAS Codes

```
/* STEP 5: Impute the missing values found in the categorical variable - GENDER */

PROC SQL;

UPDATE LIB78400.TRAINING_DS
SET GENDER = ( SELECT to.GENDER AS GENDER
               FROM LIB78400.TRAINING_GENDER_STAT_DS to
               WHERE to.COUNTS eq ( SELECT MAX(ti.COUNTS) AS HIGHEST_COUNT
                                    FROM LIB78400.TRAINING_GENDER_STAT_DS ti ) )
                                    /* Above is a sub-program to find the highest count */
WHERE ( GENDER eq '' OR GENDER IS MISSING );

QUIT;
```

8.2.6.2 Screenshot(s)/Output(s)

```
NOTE: 13 rows were updated in LIB78400.TRAINING_DS.
```

8.2.6.3 Description

The data scientist would like to impute the 13 missing values found in GENDER with its mode, male. As a result, the 13 rows with missing values for MARITAL_STATUS were updated with male gender in LIB78400.TRAINING_DS.

8.2.7 Step 6: List Out the Loan Applicants Who Did Not Detail Their Gender During the Loan Application Submission (After Imputation)

8.2.7.1 SAS Codes

```
/* STEP 6: (AI) Find the details of loan applicants who submitted their loan application without marital status */
TITLE1 'STEP 6(AI) Find the details of loan applicants who submitted';
TITLE2 'their loan application without specifying their gender';
FOOTNOTE '----------END----------';

PROC SQL;

SELECT *
FROM LIB78400.TRAINING_DS e
WHERE ( e.GENDER eq '' OR e.GENDER IS MISSING );

QUIT;
```

8.2.7.2 Screenshot(s)/Output(s)

STEP 6(AI) Find the details of loan applicants who submitted their loan application without specifying their gender

----------END----------

8.2.7.3 Description

After imputation, the data scientist would like to double-check if the details of loan applicants who did not specify their gender were still there. As a result, the details could not be found as the missing values for GENDER were imputed with ‘male’.

8.3 Imputing Missing Values Found in the Categorical Variable – EMPLOYMENT

8.3.1 Step 1: List Out the Loan Applicants Who Did Not Detail Their Employment Status During the Loan Application Submission

8.3.1.1 SAS Codes

```
643 /* STEP 1: Find the details of loan applicants who submitted their
644 loan application without specifying their employment status */
645 TITLE1 'Find the details of loan applicants who submitted their';
646 TITLE2 'loan application without specifying their employment status';
647 FOOTNOTE '----------END----------';
648
649 PROC SQL;
650
651 SELECT *
652 FROM LIB78400.TRAINING_DS e
653 WHERE ( e.EMPLOYMENT eq '' OR e.EMPLOYMENT IS MISSING );
654
655 QUIT;
```

8.3.1.2 Screenshot(s)/Output(s)

Find the details of loan applicants who submitted their
loan application without specifying their employment status

SME_LOAN_ID_NO	GENDER	MARITAL_STATUS	FAMILY_MEMBERS	QUALIFICATION	EMPLOYMENT	CANDIDATE_INCOME	GUARANTEE_INCOME	LOAN_AMOUNT	LOAN_DURATION	LOAN_HISTORY	LOAN_LOCATION	LOAN_APPROVAL_STATUS
LP001027	Male	Married	2	Graduate		2500	1840	109	360	1	City	Y
LP001041	Male	Married	0	Graduate		2600	3500	115	.	1	City	Y
LP001052	Male	Married	1	Graduate		3717	2925	151	360	.	Town	N
LP001087	Female	Not Married	2	Graduate		3750	2083	120	360	1	Town	Y
LP001091	Male	Married	1	Graduate		4166	3369	201	360	.	City	N
LP001326	Male	Not Married	0	Graduate		6782	0	.	360	.	City	N
LP001370	Male	Not Married	0	Under Graduate		7333	0	120	360	1	Village	N
LP001387	Female	Married	0	Graduate		2929	2333	139	360	1	Town	Y
LP001398	Male	Not Married	0	Graduate		5050	0	118	360	1	Town	Y
LP001546	Male	Not Married	0	Graduate		2980	2083	120	360	1	Village	Y
LP001581	Male	Married	0	Under Graduate		1820	1769	95	360	1	Village	Y
LP001732	Male	Married	2	Graduate		5000	0	72	360	0	Town	N
LP001768	Male	Married	0	Graduate		3716	0	42	180	1	Village	Y
LP001786	Male	Married	0	Graduate		5746	0	255	360	.	City	N
LP001883	Female	Not Married	0	Graduate		3418	0	135	360	1	Village	N
LP001949	Male	Married	3+	Graduate		4416	1250	110	360	1	City	Y
LP002101	Male	Married	0	Graduate		63337	0	490	180	1	City	Y
LP002110	Male	Married	1	Graduate		5250	688	160	360	1	Village	Y
LP002128	Male	Married	2	Graduate		2583	2330	125	360	1	Village	Y
LP002209	Female	Not Married	0	Graduate		2764	1459	110	360	1	City	Y
LP002226	Male	Married	0	Graduate		3333	2500	128	360	1	Town	Y
LP002237	Male	Not Married	1	Graduate		3667	0	113	180	1	City	Y
LP002319	Male	Married	0	Graduate		6256	0	160	360	.	City	Y
LP002386	Male	Not Married	0	Graduate		12876	0	405	360	1	Town	Y
LP002435	Male	Married	0	Graduate		3539	1376	55	360	1	Village	N
LP002489	Female	Not Married	1	Under Graduate		5191	0	132	360	1	Town	Y
LP002502	Female	Married	2	Under Graduate		210	2917	98	360	1	Town	Y
LP002732	Male	Not Married	0	Under Graduate		2550	2042	126	360	1	Village	Y
LP002753	Female	Not Married	1	Graduate		3652	0	95	360	1	Town	Y
LP002888	Male	Not Married	0	Graduate		3182	2917	161	360	1	City	Y
LP002949	Female	Not Married	3+	Graduate		416	41667	350	180	.	City	N
LP002950	Male	Married	0	Under Graduate		2894	2792	155	360	1	Village	Y

----------END----------

8.3.1.3 Description

The data scientist would like to discover which loan applicants did not specify their employment status during the application submission stage. There were 24 male and eight female applicants with majority of them being graduates and have positive loan records who did not specify their employment status. Overall, their monthly income ranges between $210 and $63,337 and loan duration ranges between 180 and 360 months.

8.3.2 Step 2: Count the Loan Applicants Who Did Not Detail Their Employment Status During the Loan Application Submission

8.3.2.1 SAS Codes

```
/* STEP 2: Count the number of loan applicants who submitted
their loan application without specifying their employment status */
TITLE1 'Count the number of loan applicants who submitted their loan';
TITLE2 'application without specifying their employment status';
FOOTNOTE '----------END----------';

PROC SQL;

SELECT COUNT(*) Label = 'Number of Applicants'
FROM LIB78400.TRAINING_DS e
WHERE ( e.EMPLOYMENT eq '' OR e.EMPLOYMENT IS MISSING );

QUIT;
```

8.3.2.2 Screenshot(s)/Output(s)

Count the number of loan applicants who submitted their loan application without specifying their employment status

Number of Applicants
32

----------END----------

8.3.2.3 Description

The data scientist would like to calculate the number of loan applicants who did not specify their employment status during the application submission stage. As a result, 32 applicants did not specify their employment status.

8.3.3 Step 3: Find the Statistics of Employed and Unemployed Applicants

8.3.3.1 SAS Codes

```
671 /* STEP 3: Find the statistics of employed and unemployed applicants */
672 TITLE1 'Find the statistics of employed & unemployed applicants';
673
674 PROC SQL;
675
676 SELECT e.EMPLOYMENT AS EMPLOYMENT,
677        COUNT(*) AS COUNTS
678 FROM LIB78400.TRAINING_DS e
679 WHERE ( e.EMPLOYMENT ne '' OR e.EMPLOYMENT IS NOT MISSING )
680 GROUP BY e.EMPLOYMENT;
681
682 QUIT;
```

8.3.3.2 Screenshot(s)/Output(s)

Find the statistics of employed & unemployed applicants

EMPLOYMENT	COUNTS
No	500
Yes	82

8.3.3.3 Description

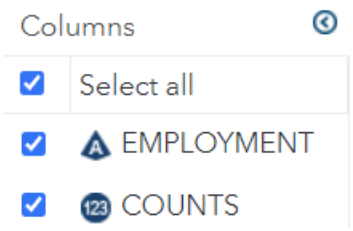
The data scientist would like to discover the statistics of employed and non-employed applicants. As a result, there were 500 non-employment applicants and 82 employed applicants.

8.3.4 Step 4: Save the Statistics of Employed and Unemployed Applicants in a Dataset

8.3.4.1 SAS Codes

```
/* STEP 4: Save the statistics in a dataset */

PROC SQL;

CREATE TABLE LIB78400.TRAINING_EMPLOYMENT_STAT_DS AS
SELECT e.EMPLOYMENT AS EMPLOYMENT,
       COUNT(*) AS COUNTS
FROM LIB78400.TRAINING_DS e
WHERE ( e.EMPLOYMENT ne '' OR e.EMPLOYMENT IS NOT MISSING )
GROUP BY e.EMPLOYMENT;

QUIT;
```

8.3.4.2 Screenshot(s)/Output(s)



Total rows: 2 Total columns: 2

	EMPLOYM...	COUNTS
1	No	500
2	Yes	82

8.3.4.3 Description

The data scientist would like to save the statistics of employed and non-employed applicants into a new dataset that would be stored in the LIB78400 library. As a result, a new dataset called LIB78400.TRAINING_EMPLOYMENT_STAT_DS was created.

8.3.5 Step 4.1: Make a Backup Copy of the Dataset Created

8.3.5.1 SAS Codes

```
/* STEP 4.1: Make a backup copy of the dataset */

PROC SQL;

CREATE TABLE LIB78400.TRAINING_BK_DS AS
SELECT *
FROM LIB78400.TRAINING_DS;

QUIT;
```

8.3.5.2 Screenshot(s)/Output(s)

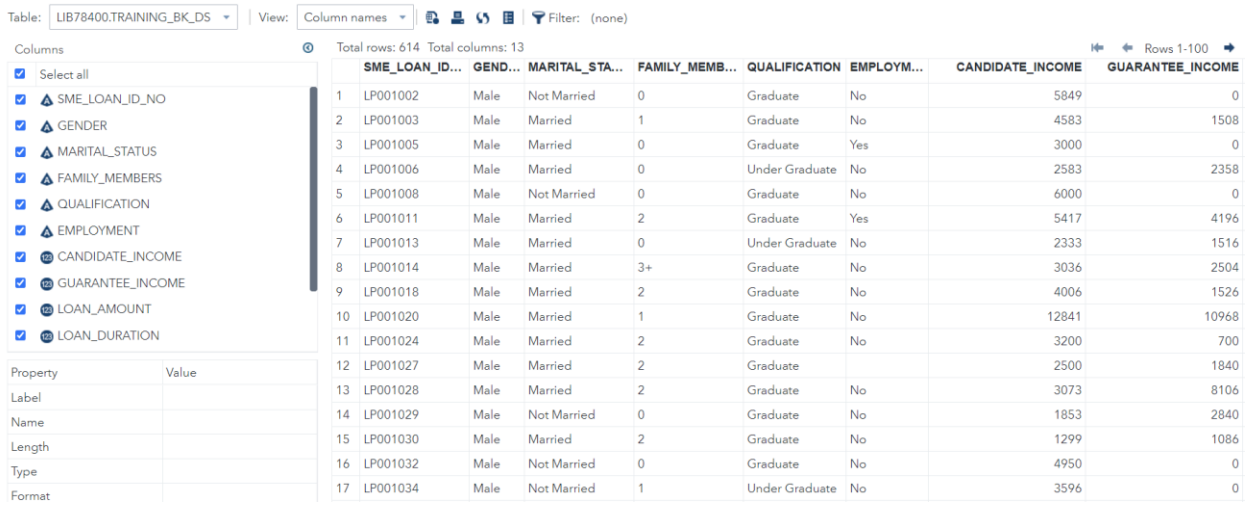


Table: LIB78400.TRAINING_BK_DS | View: Column names | Filter: (none)

Columns

- Select all
- SME_LOAN_ID_NO
- GENDER
- MARITAL_STATUS
- FAMILY_MEMBERS
- QUALIFICATION
- EMPLOYMENT
- CANDIDATE_INCOME
- GUARANTEE_INCOME
- LOAN_AMOUNT
- LOAN_DURATION

Property	Value
Label	
Name	
Length	
Type	
Format	

Total rows: 614 Total columns: 13 — Rows 1-100

	SME_LOAN_ID...	GEND...	MARITAL_STA...	FAMILY_MEMB...	QUALIFICATION	EMPLOYM...	CANDIDATE_INCOME	GUARANTEE_INCOME
1	LP001002	Male	Not Married	0	Graduate	No	5849	0
2	LP001003	Male	Married	1	Graduate	No	4583	1508
3	LP001005	Male	Married	0	Graduate	Yes	3000	0
4	LP001006	Male	Married	0	Under Graduate	No	2583	2358
5	LP001008	Male	Not Married	0	Graduate	No	6000	0
6	LP001011	Male	Married	2	Graduate	Yes	5417	4196
7	LP001013	Male	Married	0	Under Graduate	No	2333	1516
8	LP001014	Male	Married	3+	Graduate	No	3036	2504
9	LP001018	Male	Married	2	Graduate	No	4006	1526
10	LP001020	Male	Married	1	Graduate	No	12841	10968
11	LP001024	Male	Married	2	Graduate	No	3200	700
12	LP001027	Male	Married	2	Graduate		2500	1840
13	LP001028	Male	Married	2	Graduate	No	3073	8106
14	LP001029	Male	Not Married	0	Graduate	No	1853	2840
15	LP001030	Male	Married	2	Graduate	No	1299	1086
16	LP001032	Male	Not Married	0	Graduate	No	4950	0
17	LP001034	Male	Not Married	1	Under Graduate	No	3596	0

8.3.5.3 Description

The data scientist would like to make a backup copy of the training set so that he can access it whenever the original training set is corrupted during analysis. As a result, a backup copy called LIB78400.TRAINING_BK_DS was created.

8.3.6 Step 5: Impute Missing Values in the Categorical Variable – EMPLOYMENT

8.3.6.1 SAS Codes

```
/* STEP 5: Impute the missing values found in the categorical variable - EMPLOYMENT */

PROC SQL;

UPDATE LIB78400.TRAINING_DS
SET EMPLOYMENT = ( SELECT to.EMPLOYMENT AS EMPLOYMENT
                FROM LIB78400.TRAINING_EMPLOYMENT_STAT_DS to
                WHERE to.COUNTS eq ( SELECT MAX(ti.COUNTS) AS HIGHEST_COUNT
                                     FROM LIB78400.TRAINING_EMPLOYMENT_STAT_DS ti ) )
                                     /* Above is a sub-program to find the highest count */
WHERE ( EMPLOYMENT eq '' OR EMPLOYMENT IS MISSING );

QUIT;
```

8.3.6.2 Screenshot(s)/Output(s)

```
NOTE: 32 rows were updated in LIB78400.TRAINING DS.
```

8.3.6.3 Description

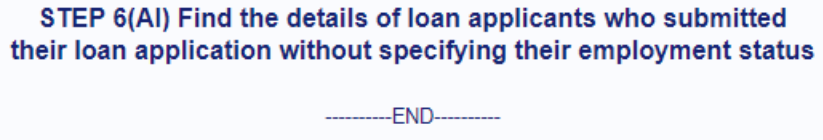
The data scientist would like to impute the 32 missing values found in EMPLOYMENT with its mode, no. As a result, the 32 rows with missing values for EMPLOYMENT were updated with 'No' in LIB78400.TRAINING_DS.

8.3.7 Step 6: List Out the Loan Applicants Who Did Not Detail Their Employment Status During the Loan Application Submission (After Imputation)

8.3.7.1 SAS Codes

```
/* STEP 6: (AI) Find the details of loan applicants who submitted their loan application without employment status */
TITLE1 'STEP 6(AI) Find the details of loan applicants who submitted';
TITLE2 'their loan application without specifying their employment status';
FOOTNOTE '----------END----------';

PROC SQL;

SELECT *
FROM LIB78400.TRAINING_DS e
WHERE ( e.EMPLOYMENT eq '' OR e.EMPLOYMENT IS MISSING );

QUIT;
```

8.3.7.2 Screenshot(s)/Output(s)



8.3.7.3 Description

After imputation, the data scientist would like to double-check if the details of loan applicants who did not specify their marital status were still there. As a result, the details could not be found as the missing values for MARITAL_STATUS were imputed with 'married'.

8.4 Imputing Missing Values Found in the Categorical Variable – FAMILY_MEMBERS

8.4.1 Step 1: List Out the Loan Applicants Who Did Not Detail Number of Family Members During the Loan Application Submission

8.4.1.1 SAS Codes

```
/* STEP 1: Find the details of loan applicants who submitted
their loan application without specifying number of family members */
TITLE1 'Find the details of loan applicants who submitted';
TITLE2 'their loan application without specifying number of family members';
FOOTNOTE '----------END----------';

PROC SQL;

SELECT *
FROM LIB78400.TRAINING_DS e
WHERE ( e.FAMILY_MEMBERS eq '' OR e.FAMILY_MEMBERS IS NULL );

QUIT;
```

8.4.1.2 Screenshot(s)/Output(s)

Find the details of loan applicants who submitted
their loan application without specifying number of family members

SME_LOAN_ID_NO	GENDER	MARITAL_STATUS	FAMILY_MEMBERS	QUALIFICATION	EMPLOYMENT	CANDIDATE_INCOME	GUARANTEE_INCOME	LOAN_AMOUNT	LOAN_DURATION	LOAN_HISTORY	LOAN_LOCATION	LOAN_APPROVAL_STATUS
LP001350	Male	Married		Graduate	No	13650	0		360	1	City	Y
LP001357	Male	Married		Graduate	No	3816	754	160	360	1	City	Y
LP001426	Male	Married		Graduate	No	5667	2667	180	360	1	Village	Y
LP001754	Male	Married		Under Graduate	Yes	4735	0	138	360	1	City	N
LP001760	Male	Married		Graduate	No	4758	0	158	480	1	Town	Y
LP001945	Female	Not Married		Graduate	No	5417	0	143	480	0	City	N
LP001972	Male	Married		Under Graduate	No	2875	1750	105	360	1	Town	Y
LP002100	Male	Not Married		Graduate	No	2833	0	71	360	1	City	Y
LP002106	Male	Married		Graduate	Yes	5503	4490	70	.	1	Town	Y
LP002130	Male	Married		Under Graduate	No	3523	3230	152	360	0	Village	N
LP002144	Female	Not Married		Graduate	No	3813	0	116	180	1	City	Y
LP002393	Female	Married		Graduate	No	10047	0	.	240	1	Town	Y
LP002682	Male	Married		Under Graduate	No	3074	1800	123	360	0	Town	N
LP002847	Male	Married		Graduate	No	5116	1451	165	360	0	City	N
LP002943	Male	Not Married		Graduate	No	2987	0	88	360	0	Town	N

----------END----------

8.4.1.3 Description

The data scientist would like to discover which loan applicants did not specify the number of family members during the application submission stage. There were 12 male and three female applicants who did not specify their marital status. Their monthly income ranges between $2,833 and $13,650 and loan duration ranges between 180 and 480 months.

8.4.2 Step 2: Count the Loan Applicants Who Did Not Detail Number of Family Members During the Loan Application Submission

8.4.2.1 SAS Codes

```
/* STEP 2: Count the number of loan applicants who submitted
their loan application without specifying number of family members */
TITLE1 'Count the number of loan applicants who submitted';
TITLE2 'their loan application without specifying number of family members';
FOOTNOTE '----------END----------';

PROC SQL;

SELECT COUNT(*) Label = 'Number of Applicants'
FROM LIB78400.TRAINING_DS e
WHERE ( e.FAMILY_MEMBERS eq '' OR e.FAMILY_MEMBERS IS MISSING );

QUIT;
```

8.4.2.2 Screenshot(s)/Output(s)

Count the number of loan applicants who submitted
their loan application without specifying number of family members

Number of Applicants
15

----------END----------

8.4.2.3 Description

The data scientist would like to calculate the number of loan applicants who did not specify their number of family members during the application submission stage. As a result, 15 applicants did not specify the number of family members.

8.4.3 Step 3: Find the Loan Applicants with Three or More Family Members

8.4.3.1 SAS Codes

```
/* STEP 3: Find the details of loan applicants with '3+'' family members */
PROC SQL;

SELECT e.FAMILY_MEMBERS Label = 'Family Members',
    SUBSTR(e.FAMILY_MEMBERS,1,1) Label = 'The data found in the 1ST position',
    SUBSTR(e.FAMILY_MEMBERS,2,1) Label = 'The data found in the 2ND position'
FROM LIB78400.TRAINING_DS e
WHERE ( e.FAMILY_MEMBERS ne '' OR e.FAMILY_MEMBERS IS NOT MISSING );

QUIT;
```

8.4.3.2 Screenshot(s)/Output(s)

Family Members	The data found in the 1ST position	The data found in the 2ND position
0	0	
1	1	
0	0	
0	0	
0	0	
2	2	
0	0	
3+	3	+
2	2	
1	1	
2	2	
2	2	
2	2	
0	0	
2	2	
0	0	
1	1	
0	0	
0	0	
0	0	
0	0	
1	1	
0	0	
2	2	
1	1	
0	0	
0	0	
2	2	
0	0	
2	2	
1	1	
0	0	
1	1	
0	0	
3+	3	+
0	0	
0	0	

8.4.3.3 Description

The data scientist would like to list the loan applicants with three or more family members. The first and second characters of each value in the FAMILY_MEMBERS column were

independently selected and placed in the two columns about data found in the first and second position respectively. As a result, for applicants whose family members had three or more, the corresponding ‘3+’ values were separated into ‘3’ and ‘+’ and each was placed in the data found in the first and second positions, respectively.

8.4.4 Step 4: Make a Backup Copy of the Dataset Created and Restore the Dataset If Corrupted

8.4.4.1 SAS Codes

```
775 /* STEP 4: Make a backup copy of the dataset */
776
777 PROC SQL;
778
779 CREATE TABLE LIB78400.TRAINING_BK_DS AS
780 SELECT *
781 FROM LIB78400.TRAINING_DS;
782
783 QUIT;
784
785 /* STEP 4: To restore the dataset if it's corrupted */
786
787 PROC SQL;
788
789 CREATE TABLE LIB78400.TRAINING_DS AS
790 SELECT *
791 FROM LIB78400.TRAINING_BK_DS;
792
793 QUIT;
```

8.4.4.2 Screenshot(s)/Output(s)

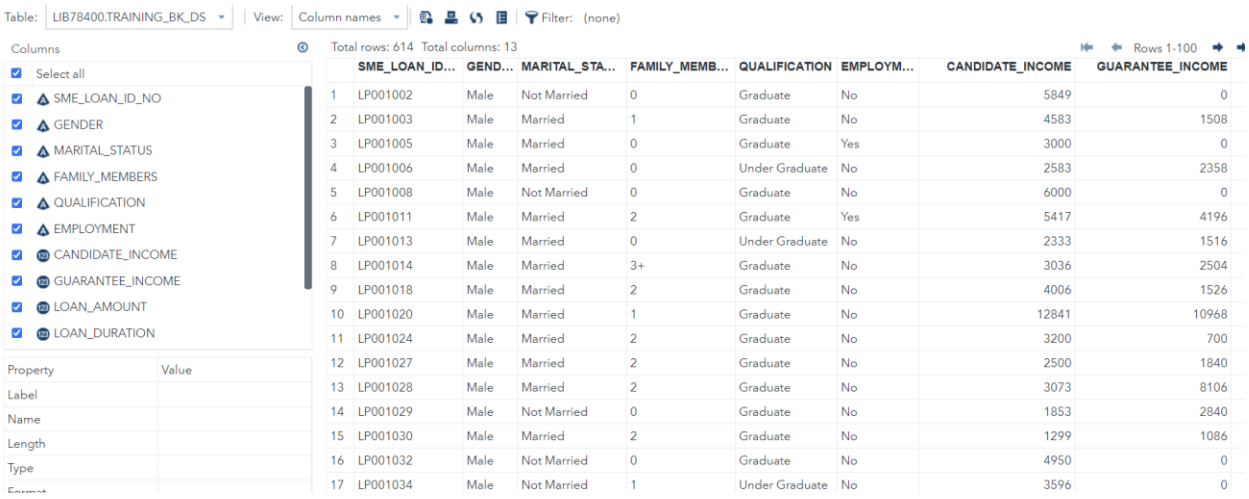


Table: LIB78400.TRAINING_BK_DS | View: Column names | Filter: (none)

Columns

- Select all
- SME_LOAN_ID_NO
- GENDER
- MARITAL_STATUS
- FAMILY_MEMBERS
- QUALIFICATION
- EMPLOYMENT
- CANDIDATE_INCOME
- GUARANTEE_INCOME
- LOAN_AMOUNT
- LOAN_DURATION

Property	Value
Label	
Name	
Length	
Type	

Total rows: 614 Total columns: 13 — Rows 1-100

	SME_LOAN_ID...	GEND...	MARITAL_STA...	FAMILY_MEMB...	QUALIFICATION	EMPLOYM...	CANDIDATE_INCOME	GUARANTEE_INCOME
1	LP001002	Male	Not Married	0	Graduate	No	5849	0
2	LP001003	Male	Married	1	Graduate	No	4583	1508
3	LP001005	Male	Married	0	Graduate	Yes	3000	0
4	LP001006	Male	Married	0	Under Graduate	No	2583	2358
5	LP001008	Male	Not Married	0	Graduate	No	6000	0
6	LP001011	Male	Married	2	Graduate	Yes	5417	4196
7	LP001013	Male	Married	0	Under Graduate	No	2333	1516
8	LP001014	Male	Married	3+	Graduate	No	3036	2504
9	LP001018	Male	Married	2	Graduate	No	4006	1526
10	LP001020	Male	Married	1	Graduate	No	12841	10968
11	LP001024	Male	Married	2	Graduate	No	3200	700
12	LP001027	Male	Married	2	Graduate	No	2500	1840
13	LP001028	Male	Married	2	Graduate	No	3073	8106
14	LP001029	Male	Not Married	0	Graduate	No	1853	2840
15	LP001030	Male	Married	2	Graduate	No	1299	1086
16	LP001032	Male	Not Married	0	Graduate	No	4950	0
17	LP001034	Male	Not Married	1	Under Graduate	No	3596	0

8.4.4.3 Description

The data scientist would like to make a backup copy of the training set so that he can access it whenever the original training set is corrupted during analysis. As a result, a backup copy called LIB78400.TRAINING_BK_DS was created.

8.4.5 Step 5: Remove '+' in the Values of FAMILY_MEMBERS

8.4.5.1 SAS Codes

```
795 /* STEP 5: Remove '+' symbol found in FAMILY_MEMBERS */
796 PROC SQL;
797
798 UPDATE LIB78400.TRAINING_DS
799 SET FAMILY_MEMBERS = SUBSTR(FAMILY_MEMBERS,1,1)
800 WHERE SUBSTR(FAMILY_MEMBERS,2,1) eq '+';
801
802 QUIT;
```

8.4.5.2 Screenshot(s)/Output(s)

```
NOTE: 51 rows were updated in LIB78400.TRAINING_DS.
```

8.4.5.3 Description

The data scientist would like to remove the '+' symbol from each of the '3+' values in FAMILY_MEMBERS. As a result, the '+' symbols were deleted and each of the '3+' values in those 51 rows had the '+' symbol removed, remaining the '3' values only.

8.4.6 Step 6: Find the Statistics of Loan Applicants Based on the Number of Family Members

8.4.6.1 SAS Codes

```
804 /* STEP 6: Display the statistics in a dataset */
805
806 PROC SQL;
807
808 SELECT e.FAMILY_MEMBERS AS FAMILY_MEMBERS,
809        COUNT(*) AS COUNTS
810 FROM LIB78400.TRAINING_DS e
811 WHERE ( e.FAMILY_MEMBERS ne '' OR e.FAMILY_MEMBERS IS NOT MISSING )
812 GROUP BY e.FAMILY_MEMBERS;
813
814 QUIT;
```

8.4.6.2 Screenshot(s)/Output(s)

FAMILY_MEMBERS	COUNTS
0	345
1	102
2	101
3	51

8.4.6.3 Description

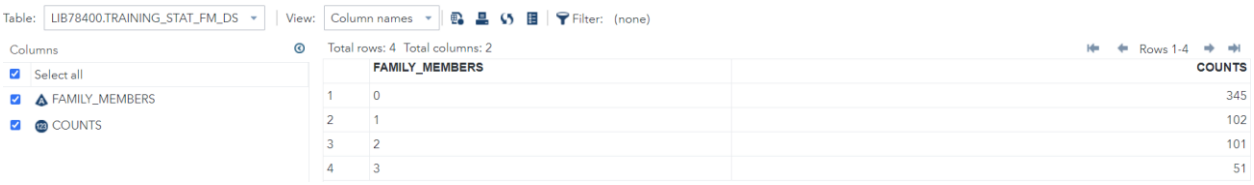
The data scientist would like to discover the statistics of loan applicants regarding the number of family members. As a result, there were 345 applicants with no family members, 102 with one family member, 101 with two family members and 51 with three family members.

8.4.7 Step 7: Save the Statistics of Married and Non-married Loan Applicants in a Dataset

8.4.7.1 SAS Codes

```
816 /* STEP 7: Save the statistics in a dataset */
817
818 PROC SQL;
819
820 CREATE TABLE LIB78400.TRAINING_STAT_FM_DS AS
821 SELECT e.FAMILY_MEMBERS AS FAMILY_MEMBERS,
822        COUNT(*) AS COUNTS
823 FROM LIB78400.TRAINING_DS e
824 WHERE ( e.FAMILY_MEMBERS ne '' OR e.FAMILY_MEMBERS IS NOT MISSING )
825 GROUP BY e.FAMILY_MEMBERS;
826
827 QUIT;
```

8.4.7.2 Screenshot(s)/Output(s)



8.4.7.3 Description

The data scientist would like to save the statistics of married and non-married applicants into a new dataset stored in the LIB78400 library. As a result, a new dataset called LIB78400.TRAINING_STAT_FM_DS was created.

8.4.8 Step 8: Impute Missing Values in the Categorical Variable – FAMILY_MEMBERS

8.4.8.1 SAS Codes

```
/* STEP 8: Impute missing values in categorical variable - FAMILY_MEMBERS */

PROC SQL;

UPDATE LIB78400.TRAINING_DS
SET FAMILY_MEMBERS =  ( SELECT to.FAMILY_MEMBERS AS FAMILY_MEMBERS
                        FROM LIB78400.TRAINING_STAT_FM_DS to
                        WHERE to.COUNTS eq ( SELECT MAX (ti.COUNTS) AS HIGHEST_COUNT
                                             FROM LIB78400.TRAINING_STAT_FM_DS ti ) )
                                             /* Above is a sub-program to find the highest count */
WHERE ( FAMILY_MEMBERS eq '' OR FAMILY_MEMBERS IS NULL );

QUIT;
```

8.4.8.2 Screenshot(s)/Output(s)

```
NOTE: 15 rows were updated in LIB78400.TRAINING_DS.
```

8.4.8.3 Description

The data scientist would like to impute the 15 missing values found in FAMILY_MEMBERS with its mode, ‘0’. As a result, the 15 rows with missing values for FAMILY_MEMBERS were updated with ‘0’ values in LIB78400.TRAINING_DS.

8.4.9 Step 9: List Out the Loan Applicants Who Did Not Detail Number of Family Members During the Loan Application Submission (After Imputation)

8.4.9.1 SAS Codes

```
/* STEP 9: (AI) Find the details of loan applicants who submitted
their loan application without specifying number of family members */
TITLE1 'Find the details of loan applicants who submitted';
TITLE2 'their loan application without specifying number of family members';
FOOTNOTE '----------END----------';

PROC SQL;

SELECT *
FROM LIB78400.TRAINING_DS e
WHERE ( e.FAMILY_MEMBERS eq '' OR e.FAMILY_MEMBERS IS NULL );

QUIT;
```

8.4.9.2 Screenshot(s)/Output(s)

Find the details of loan applicants who submitted
their loan application without specifying number of family members

---------END---------

8.4.9.3 Description

After imputation, the data scientist would like to double-check if the details of loan applicants who did not specify the number of family members were still there. As a result, the details could not be found as the missing values for FAMILY_MEMBERS were imputed with '0'.

8.4.10 Step 10: Count the Loan Applicants Who Did Not Detail Number of Family Members During the Loan Application Submission (After Imputation)

8.4.10.1 SAS Codes

```
857 /* STEP 10: (AI) Count the number of loan applicants who submitted
858 their loan application without specifying number of family members */
859 TITLE1 'Count the number of loan applicants who submitted';
860 TITLE2 'their loan application without specifying number of family members';
861 FOOTNOTE '----------END----------';
862
863 PROC SQL;
864
865 SELECT COUNT(*) Label = 'Number of Applicants'
866 FROM LIB78400.TRAINING_DS e
867 WHERE ( e.FAMILY_MEMBERS eq '' OR e.FAMILY_MEMBERS IS MISSING );
868
869 QUIT;
```

8.4.10.2 Screenshot(s)/Output(s)

Count the number of loan applicants who submitted
their loan application without specifying number of family members

Number of Applicants
0

---------END---------

8.4.10.3 Description

After imputation, the data scientist would like to double-check if the number of loan applicants who did not specify the number of family members can still be calculated. As a result, no applicants were found due to the imputation of '0'.

9.0 Imputing Missing Values Found in the Continuous Variables of LIB78400.TRAINING_DS

9.1 Imputing Missing Values Found in the Continuous Variable – LOAN_AMOUNT

9.1.1 Step 1: List Out the Loan Applicants Who Did Not Detail Their Marital Status During the Loan Application Submission

9.1.1.1 SAS Codes

```
966 /* STEP 1: Find the details of loan applicants who submitted their
967 loan application without specifying loan amount */
968 TITLE1 'Find the details of loan applicants who submitted their';
969 TITLE2 'loan application without specifying loan amount';
970 FOOTNOTE '----------END----------';
971
972 PROC SQL;
973
974 SELECT *
975 FROM LIB78400.TRAINING_DS e
976 WHERE ( e.LOAN_AMOUNT eq . OR e.LOAN_AMOUNT IS MISSING );
977
978 QUIT;
```

9.1.1.2 Screenshot(s)/Output(s)

Find the details of loan applicants who submitted their
loan application without specifying loan amount

SME_LOAN_ID_NO	GENDER	MARITAL_STATUS	FAMILY_MEMBERS	QUALIFICATION	EMPLOYMENT	CANDIDATE_INCOME	GUARANTEE_INCOME	LOAN_AMOUNT	LOAN_DURATION	LOAN_HISTORY	LOAN_LOCATION	LOAN_APPROVAL_STATUS
LP001002	Male	Not Married	0	Graduate	No	5849	0	.	360	1	City	Y
LP001106	Male	Married	0	Graduate	No	2275	2067	.	360	1	City	Y
LP001213	Male	Married	1	Graduate	No	4945	0	.	360	0	Village	N
LP001266	Male	Married	1	Graduate	Yes	2395	0	.	360	1	Town	Y
LP001326	Male	Not Married	0	Graduate	No	6782	0	.	360	1	City	N
LP001350	Male	Married	0	Graduate	No	13650	0	.	360	1	City	Y
LP001356	Male	Married	0	Graduate	No	4652	3583	.	360	1	Town	Y
LP001392	Female	Not Married	1	Graduate	Yes	7451	0	.	360	1	Town	Y
LP001449	Male	Not Married	0	Graduate	No	3865	1640	.	360	1	Village	Y
LP001682	Male	Married	3	Under Graduate	No	3992	0	.	180	1	City	N
LP001922	Male	Married	0	Graduate	No	20667	0	.	360	1	Village	N
LP001990	Male	Not Married	0	Under Graduate	No	2000	0	.	360	1	City	N
LP002054	Male	Married	2	Under Graduate	No	3601	1590	.	360	1	Village	Y
LP002113	Female	Not Married	3	Under Graduate	No	1830	0	.	360	0	City	N
LP002243	Male	Married	0	Under Graduate	No	3010	3136	.	360	0	City	N
LP002393	Female	Married	0	Graduate	No	10047	0	.	240	1	Town	Y
LP002401	Male	Married	0	Graduate	No	2213	1125	.	360	1	City	Y
LP002533	Male	Married	2	Graduate	No	2947	1603	.	360	1	City	N
LP002697	Male	Not Married	0	Graduate	No	4680	2087	.	360	1	Town	N
LP002778	Male	Married	2	Graduate	Yes	6633	0	.	360	0	Village	N
LP002784	Male	Married	1	Under Graduate	No	2492	2375	.	360	1	Village	Y
LP002960	Male	Married	0	Under Graduate	No	2400	3800	.	180	1	City	N

----------END----------

9.1.1.3 Description

The data scientist would like to discover which loan applicants did not specify the loan amount during the application submission stage. 19 male and three female applicants who were mostly married, graduates, unemployed and had positive loan records did not specify their loan amount. Their monthly income ranges between $1,830 and $20,667 and loan duration ranges between 180 and 360 months.

9.1.2 Step 2: Count the Loan Applicants Who Did Not Detail Their Loan Amount During the Loan Application Submission

9.1.2.1 SAS Codes

```
/* STEP 2: Count the number of loan applicants who submitted
their loan application without specifying loan amount */
TITLE1 'Count the number of loan applicants who submitted their loan'
TITLE2 'application without specifying loan amount';
FOOTNOTE '----------END----------';

PROC SQL;

SELECT COUNT(*) Label = 'Number of Applicants'
FROM LIB78400.TRAINING_DS e
WHERE ( e.LOAN_AMOUNT eq . OR e.LOAN_AMOUNT IS MISSING );

QUIT;
```

9.1.2.2 Screenshot(s)/Output(s)

Count the number of loan applicants who submitted their loan application without specifying loan amount

Number of Applicants
22

----------END----------

9.1.2.3 Description

The data scientist would like to calculate the number of loan applicants who did not specify their loan amount during the application submission stage. As a result, 22 applicants did not specify their desired loan amount.

9.1.3 Step 3: Impute Missing Values in the Continuous Variable – LOAN_AMOUNT

9.1.3.1 SAS Codes

```
/* STEP 3: Impute the missing values found in the continuous variable - LOAN_AMOUNT */

PROC STDIZE DATA = LIB78400.TRAINING_DS REPONLY

METHOD = MEAN OUT = LIB78400.TRAINING_DS;
VAR LOAN_AMOUNT;

QUIT;
```

9.1.3.2 Screenshot(s)/Output(s)

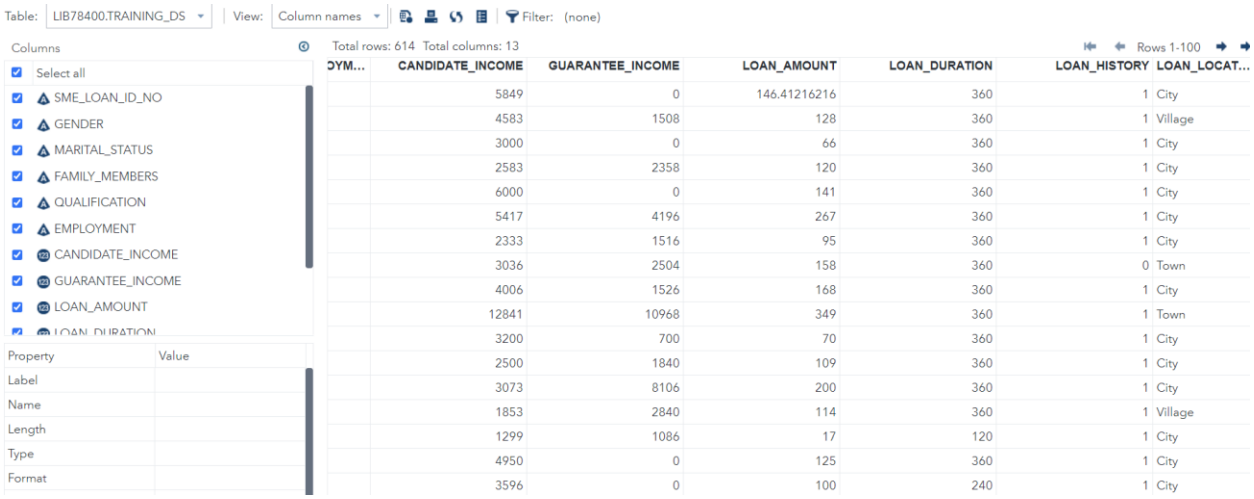


Table: LIB78400.TRAINING_DS | View: Column names | Filter: (none)

Columns: Select all, SME_LOAN_ID_NO, GENDER, MARITAL_STATUS, FAMILY_MEMBERS, QUALIFICATION, EMPLOYMENT, CANDIDATE_INCOME, GUARANTEE_INCOME, LOAN_AMOUNT, LOAN_DURATION

Property	Value
Label	
Name	
Length	
Type	
Format	

Total rows: 614 Total columns: 13 — Rows 1-100

...OYM...	CANDIDATE_INCOME	GUARANTEE_INCOME	LOAN_AMOUNT	LOAN_DURATION	LOAN_HISTORY	LOAN_LOCAT...
	5849	0	146.41216216	360	1	City
	4583	1508	128	360	1	Village
	3000	0	66	360	1	City
	2583	2358	120	360	1	City
	6000	0	141	360	1	City
	5417	4196	267	360	1	City
	2333	1516	95	360	1	City
	3036	2504	158	360	0	Town
	4006	1526	168	360	1	City
	12841	10968	349	360	1	Town
	3200	700	70	360	1	City
	2500	1840	109	360	1	City
	3073	8106	200	360	1	City
	1853	2840	114	360	1	Village
	1299	1086	17	120	1	City
	4950	0	125	360	1	City
	3596	0	100	240	1	City

9.1.3.3 Description

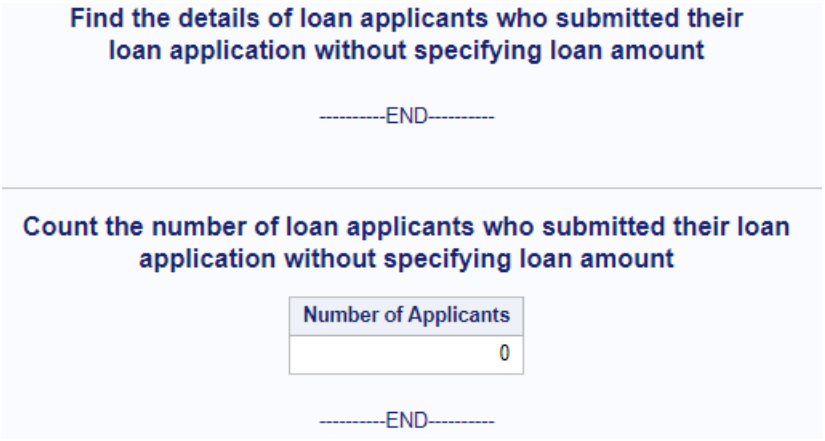
The data scientist would like to impute the 22 missing values found in LOAN_AMOUNT with its average. As a result, the 22 rows with missing values for LOAN_AMOUNT were imputed with the mean value of LOAN_AMOUNT in LIB78400.TRAINING_DS.

9.1.4 Step 4 & 5: List Out and Count the Loan Applicants Who Did Not Detail Their Loan Amount During the Loan Application Submission (After Imputation)

9.1.4.1 SAS Codes

```
/* STEP 4: (AI) Find the details of loan applicants who submitted their
loan application without specifying loan amount */
TITLE1 'Find the details of loan applicants who submitted their';
TITLE2 'loan application without specifying loan amount';
FOOTNOTE '----------END----------';

PROC SQL;

SELECT *
FROM LIB78400.TRAINING_DS e
WHERE ( e.LOAN_AMOUNT eq . OR e.LOAN_AMOUNT IS MISSING );

QUIT;

/* STEP 5: (AI) Count the number of loan applicants who submitted
their loan application without specifying loan amount */
TITLE1 'Count the number of loan applicants who submitted their loan';
TITLE2 'application without specifying loan amount';
FOOTNOTE '----------END----------';

PROC SQL;

SELECT COUNT(*) Label = 'Number of Applicants'
FROM LIB78400.TRAINING_DS e
WHERE ( e.LOAN_AMOUNT eq . OR e.LOAN_AMOUNT IS MISSING );

QUIT;
```

9.1.4.2 Screenshot(s)/Output(s)



Find the details of loan applicants who submitted their loan application without specifying loan amount

----------END----------

Count the number of loan applicants who submitted their loan application without specifying loan amount

Number of Applicants
0

----------END----------

9.1.4.3 Description

After imputation, the data scientist would like to double-check if the details of loan applicants who did not specify their loan amount were still there. As a result, the details and number of applicants could not be found as the missing values for LOAN_AMOUNT were imputed with the mean of LOAN_AMOUNT.

9.2 Imputing Missing Values Found in the Continuous Variable – LOAN_DURATION

9.2.1 Step 1: List Out the Loan Applicants Who Did Not Detail Loan Duration During the Loan Application Submission

9.2.1.1 SAS Codes

```
1043 /* STEP 1: Find the details of loan applicants who submitted their
1044 loan application without specifying loan duration */
1045 TITLE1 'Find the details of loan applicants who submitted their';
1046 TITLE2 'loan application without specifying loan duration';
1047 FOOTNOTE '----------END----------';
1048
1049 PROC SQL;
1050
1051 SELECT *
1052 FROM LIB78400.TRAINING_DS e
1053 WHERE ( e.LOAN_DURATION eq . OR e.LOAN_DURATION IS MISSING );
1054
1055 QUIT;
```

9.2.1.2 Screenshot(s)/Output(s)

Find the details of loan applicants who submitted their loan application without specifying loan duration

SME_LOAN_ID_NO	GENDER	MARITAL_STATUS	FAMILY_MEMBERS	QUALIFICATION	EMPLOYMENT	CANDIDATE_INCOME	GUARANTEE_INCOME	LOAN_AMOUNT	LOAN_DURATION	LOAN_HISTORY	LOAN_LOCATION	LOAN_APPROVAL_STATUS
LP001041	Male	Married	0	Graduate	No	2600	3500	115	.	1	City	Y
LP001109	Male	Married	0	Graduate	No	1828	1330	100	.	0	City	N
LP001136	Male	Married	0	Under Graduate	Yes	4695	0	96	.	1	City	Y
LP001137	Female	Not Married	0	Graduate	No	3410	0	88	.	1	City	Y
LP001250	Male	Married	3	Under Graduate	No	4755	0	95	.	0	Town	N
LP001391	Male	Married	0	Under Graduate	No	3572	4114	152	.	0	Village	N
LP001574	Male	Married	0	Graduate	No	3707	3166	182	.	1	Village	Y
LP001669	Female	Not Married	0	Under Graduate	No	1907	2365	120	.	1	City	Y
LP001749	Male	Married	0	Graduate	No	7578	1010	175	.	1	Town	Y
LP001770	Male	Not Married	0	Under Graduate	No	3189	2598	120	.	1	Village	Y
LP002106	Male	Married	0	Graduate	Yes	5503	4490	70	.	1	Town	Y
LP002188	Male	Not Married	0	Graduate	No	5124	0	124	.	0	Village	N
LP002357	Female	Not Married	0	Under Graduate	No	2720	0	80	.	0	City	N
LP002362	Male	Married	1	Graduate	No	7250	1667	110	.	0	City	N

----------END----------

9.2.1.3 Description

The data scientist would like to discover which loan applicants did not specify the loan duration during the application submission stage. 11 male and three female applicants who were mostly unemployed and had positive loan records did not specify their loan duration. Their monthly income ranges between $1,828 and $7,578.

9.2.2 Step 2: Count the Loan Applicants Who Did Not Detail Loan Duration During the Loan Application Submission

9.2.2.1 SAS Codes

```
1057 /* STEP 2: Count the number of loan applicants who submitted
1058 their loan application without specifying loan duration */
1059 TITLE1 'Count the number of loan applicants who submitted their loan';
1060 TITLE2 'application without specifying loan duration';
1061 FOOTNOTE '----------END----------';
1062
1063 PROC SQL;
1064
1065 SELECT COUNT(*) Label = 'Number of Applicants'
1066 FROM LIB78400.TRAINING_DS e
1067 WHERE ( e.LOAN_DURATION eq . OR e.LOAN_DURATION IS MISSING );
1068
1069 QUIT;
```

9.2.2.2 Screenshot(s)/Output(s)

Count the number of loan applicants who submitted their loan application without specifying loan duration

Number of Applicants
14

----------END----------

9.2.2.3 Description

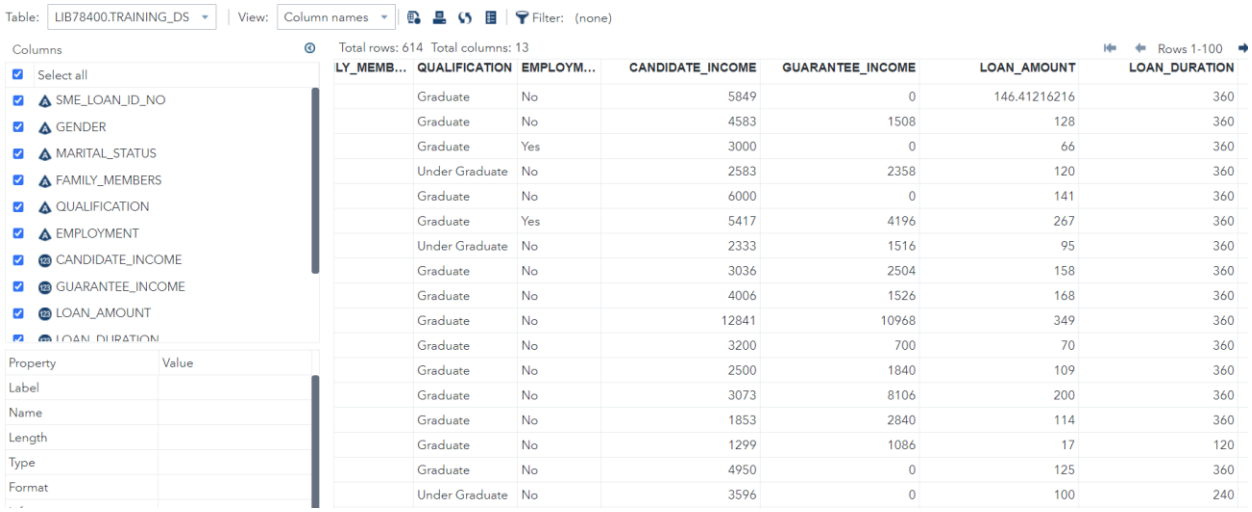
The data scientist would like to calculate the number of loan applicants who did not specify their loan amount during the application submission stage. As a result, 14 applicants did not specify their loan amount.

9.2.3 Step 3: Impute Missing Values in the Categorical Variable – LOAN_DURATION

9.2.3.1 SAS Codes

```
/* STEP 3: Impute the missing values found in the continuous variable - LOAN_DURATION */

PROC STDIZE DATA = LIB78400.TRAINING_DS REPONLY

METHOD = MEAN OUT = LIB78400.TRAINING_DS;
VAR LOAN_DURATION;

QUIT;
```

9.2.3.2 Screenshot(s)/Output(s)



LY_MEMB...	QUALIFICATION	EMPLOYM...	CANDIDATE_INCOME	GUARANTEE_INCOME	LOAN_AMOUNT	LOAN_DURATION
	Graduate	No	5849	0	146.41216216	360
	Graduate	No	4583	1508	128	360
	Graduate	Yes	3000	0	66	360
	Under Graduate	No	2583	2358	120	360
	Graduate	No	6000	0	141	360
	Graduate	Yes	5417	4196	267	360
	Under Graduate	No	2333	1516	95	360
	Graduate	No	3036	2504	158	360
	Graduate	No	4006	1526	168	360
	Graduate	No	12841	10968	349	360
	Graduate	No	3200	700	70	360
	Graduate	No	2500	1840	109	360
	Graduate	No	3073	8106	200	360
	Graduate	No	1853	2840	114	360
	Graduate	No	1299	1086	17	120
	Graduate	No	4950	0	125	360
	Under Graduate	No	3596	0	100	240

9.2.3.3 Description

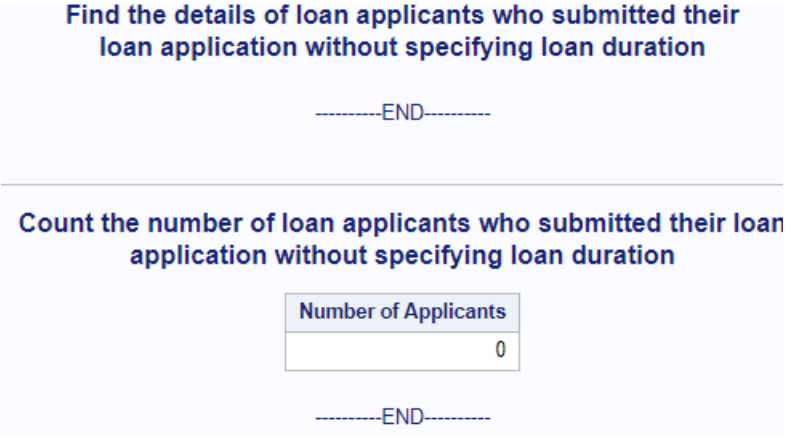
The data scientist would like to impute the 14 missing values found in LOAN_DURATION with its average. As a result, the 14 rows with missing values for LOAN_DURATION were imputed with the mean value of LOAN_DURATION in LIB78400.TRAINING_DS.

9.2.4 Step 4 & 5: List Out and Count the Loan Applicants Who Did Not Detail Loan Duration During the Loan Application Submission (After Imputation)

9.2.4.1 SAS Codes

```
/* STEP 4: (AI) Find the details of loan applicants who submitted their
loan application without specifying loan duration */
TITLE1 'Find the details of loan applicants who submitted their';
TITLE2 'loan application without specifying loan duration';
FOOTNOTE '----------END----------';

PROC SQL;

SELECT *
FROM LIB78400.TRAINING_DS e
WHERE ( e.LOAN_DURATION eq . OR e.LOAN_DURATION IS MISSING );

QUIT;

/* STEP 5: (AI) Count the number of loan applicants who submitted
their loan application without specifying loan duration */
TITLE1 'Count the number of loan applicants who submitted their loan';
TITLE2 'application without specifying loan duration';
FOOTNOTE '----------END----------';

PROC SQL;

SELECT COUNT(*) Label = 'Number of Applicants'
FROM LIB78400.TRAINING_DS e
WHERE ( e.LOAN_DURATION eq . OR e.LOAN_DURATION IS MISSING );

QUIT;
```

9.2.4.2 Screenshot(s)/Output(s)



Find the details of loan applicants who submitted their loan application without specifying loan duration

----------END----------

Count the number of loan applicants who submitted their loan application without specifying loan duration

Number of Applicants
0

----------END----------

9.2.4.3 Description

After imputation, the data scientist would like to double-check if the details of loan applicants who did not specify their loan duration were still there. As a result, the details and number of applicants could not be found as the missing values for LOAN_DURATION were imputed with the mean value of LOAN_DURATION.

10.0 Imputing Missing Values Found in the Categorical Variables of LIB78400.TESTING_DS

10.1 Imputing Missing Values Found in the Categorical Variable – GENDER

10.1.1 Step 1: List Out the Loan Applicants Who Did Not Detail Their Gender During the Loan Application Submission

10.1.1.1 SAS Codes

```
1122 /* STEP 1: Find the details of loan applicants who submitted
1123 their loan application without specifying their gender */
1124 TITLE1 'Find the details of loan applicants who submitted';
1125 TITLE2 'their loan application without specifying their gender';
1126 FOOTNOTE '----------END----------';
1127
1128 PROC SQL;
1129
1130 SELECT *
1131 FROM LIB78400.TESTING_DS e
1132 WHERE ( e.GENDER eq '' OR e.GENDER IS MISSING );
1133
1134 QUIT;
```

10.1.1.2 Screenshot(s)/Output(s)

Find the details of loan applicants who submitted
their loan application without specifying their gender

SME_LOAN_ID_NO	GENDER	MARITAL_STATUS	FAMILY_MEMBERS	QUALIFICATION	EMPLOYMENT	CANDIDATE_INCOME	GUARANTEE_INCOME	LOAN_AMOUNT	LOAN_DURATION	LOAN_HISTORY	LOAN_LOCATION	LOAN_APPROVAL_STATUS
LP001128		Not Married	0	Graduate	No	3909	0	101	360	1	City	
LP001287		Married	3+	Under Graduate	No	3500	833	120	360	1	Town	
LP001563		Not Married	0	Graduate	No	1596	1760	119	360	0	City	
LP001769		Not Married		Graduate	No	3333	1250	110	360	1	Town	
LP002165		Not Married	1	Under Graduate	No	2038	4027	100	360	1	Village	
LP002298		Not Married	0	Graduate	Yes	2860	2988	138	360	1	City	
LP002355		Married	0	Graduate	No	3186	3145	150	180	0	Town	
LP002553		Not Married	0	Graduate	No	29167	0	185	360	1	Town	
LP002614		Not Married	0	Graduate	No	6478	0	108	360	1	Town	
LP002657		Married	1	Under Graduate	Yes	570	2125	68	360	1	Village	
LP002775		Not Married	0	Under Graduate	No	4768	0	125	360	1	Village	

----------END----------

10.1.1.3 Description

In the testing set, the data scientist would like to discover which loan applicants did not specify their gender during the application submission stage. Three married and eight non-married applicants who were mostly graduates, non-employed and had positive loan records did not specify their gender.

10.1.2 Step 2: Count the Loan Applicants Who Did Not Detail Their Gender During the Loan Application Submission

10.1.2.1 SAS Codes

```
1136 /* STEP 2: Count the number of loan applicants who submitted
1137 their loan application without specifying their gender */
1138 TITLE1 'Count the number of loan applicants who submitted';
1139 TITLE2 'their loan application without without specifying their gender';
1140 FOOTNOTE '----------END----------';
1141
1142 PROC SQL;
1143
1144 SELECT COUNT(*) Label = 'Number of Applicants'
1145 FROM LIB78400.TESTING_DS e
1146 WHERE ( e.GENDER eq '' OR e.GENDER IS MISSING );
1147
1148 QUIT;
```

10.1.2.2 Screenshot(s)/Output(s)


Count the number of loan applicants who submitted
their loan application without without specifying their gender

Number of Applicants
11

----------END----------

10.1.2.3 Description

The data scientist would like to calculate the number of loan applicants who did not specify their gender during the application submission stage. As a result, 11 applicants did not specify their gender.

10.1.3 Step 3: Find the Statistics of Male and Female Loan Applicants

10.1.3.1 SAS Codes

```
1150 /* STEP 3: Find the statistics of male and female applicants */
1151 TITLE1 'Find the statistics of male & female applicants';
1152
1153 PROC SQL;
1154
1155 SELECT e.GENDER AS GENDER,
1156        COUNT(*) AS COUNTS
1157 FROM LIB78400.TESTING_DS e
1158 WHERE ( e.GENDER ne '' OR e.GENDER IS NOT MISSING )
1159 GROUP BY e.GENDER;
1160
1161 QUIT;
```

10.1.3.2 Screenshot(s)/Output(s)

Find the statistics of male & female applicants

GENDER	COUNTS
Female	70
Male	286

10.1.3.3 Description

The data scientist would like to discover the statistics of male and female applicants. As a result, there were 286 male applicants and 70 female applicants.

10.1.4 Step 4: Save the Statistics of Male and Female Loan Applicants in a Dataset

10.1.4.1 SAS Codes

```
1163 /* STEP 4: Save the statistics in a dataset */
1164
1165 PROC SQL;
1166
1167 CREATE TABLE LIB78400.TESTING_GENDER_STAT_DS AS
1168 SELECT e.GENDER AS GENDER,
1169        COUNT(*) AS COUNTS
1170 FROM LIB78400.TESTING_DS e
1171 WHERE ( e.GENDER ne '' OR e.GENDER IS NOT MISSING )
1172 GROUP BY e.GENDER;
1173
1174 QUIT;
```

10.1.4.2 Screenshot(s)/Output(s)

Table: LIB78400.TESTING_GENDER_STAT_DS | View: Column names

Columns

- Select all
- GENDER
- COUNTS

Total rows: 2 Total columns: 2

	GENDER	COUNTS
1	Female	70
2	Male	286

10.1.4.3 Description

The data scientist would like to save the statistics of male and female applicants into a new dataset that would be stored in the LIB78400 library. As a result, a new dataset called LIB78400.TESTING_GENDER_STAT_DS was created.

10.1.5 Step 4.1: Make a Backup Copy of the Dataset Created

10.1.5.1 SAS Codes

```
1176 /* STEP 4.1: Make a backup copy of the dataset */
1177
1178 PROC SQL;
1179
1180 CREATE TABLE LIB78400.TESTING_BK_DS AS
1181 SELECT *
1182 FROM LIB78400.TESTING_DS;
1183
1184 QUIT;
```

10.1.5.2 Screenshot(s)/Output(s)

Table: LIB78400.TESTING_BK_DS | View: Column names | Filter: (none)

Columns

- Select all
- SME_LOAN_ID_NO
- GENDER
- MARITAL_STATUS
- FAMILY_MEMBERS
- QUALIFICATION
- EMPLOYMENT
- CANDIDATE_INCOME
- GUARANTEE_INCOME
- LOAN_AMOUNT
- LOAN_DURATION

Property	Value
Label	
Name	
Length	
Type	
Format	

Total rows: 367 Total columns: 13 Rows 1-100

	SME_LOAN_ID...	GEND...	MARITAL_STA...	FAMILY_MEMB...	QUALIFICATION	EMPLOYM...	CANDIDATE_INCOME	GUARANTEE_INCOME
1	LP001015	Male	Married	0	Graduate	No	5720	0
2	LP001022	Male	Married	1	Graduate	No	3076	1500
3	LP001031	Male	Married	2	Graduate	No	5000	1800
4	LP001035	Male	Married	2	Graduate	No	2340	2546
5	LP001051	Male	Not Married	0	Under Graduate	No	3276	0
6	LP001054	Male	Married	0	Under Graduate	Yes	2165	3422
7	LP001055	Female	Not Married	1	Under Graduate	No	2226	0
8	LP001056	Male	Married	2	Under Graduate	No	3881	0
9	LP001059	Male	Married	2	Graduate		13633	0
10	LP001067	Male	Not Married	0	Under Graduate	No	2400	2400
11	LP001078	Male	Not Married	0	Under Graduate	No	3091	0
12	LP001082	Male	Married	1	Graduate		2185	1516
13	LP001083	Male	Not Married	3+	Graduate	No	4166	0
14	LP001094	Male	Married	2	Graduate		12173	0
15	LP001096	Female	Not Married	0	Graduate	No	4666	0
16	LP001099	Male	Not Married	1	Graduate	No	5667	0
17	LP001105	Male	Married	2	Graduate	No	4583	2916

10.1.5.3 Description

The data scientist would like to make a backup copy of the testing set so that he can access it whenever the original testing set is corrupted during analysis. As a result, a backup copy called LIB78400.TESTING_BK_DS was created.

10.1.6 Step 5: Impute Missing Values in the Categorical Variable – GENDER

10.1.6.1 SAS Codes

```
/* STEP 5: Impute the missing values found in the categorical variable - GENDER */

PROC SQL;

UPDATE LIB78400.TESTING_DS
SET GENDER = ( SELECT to.GENDER AS GENDER
               FROM LIB78400.TESTING_GENDER_STAT_DS to
               WHERE to.COUNTS eq ( SELECT MAX(ti.COUNTS) AS HIGHEST_COUNT
                                    FROM LIB78400.TESTING_GENDER_STAT_DS ti ) )
                                    /* Above is a sub-program to find the highest count */
WHERE ( GENDER eq '' OR GENDER IS MISSING );

QUIT;
```

10.1.6.2 Screenshot(s)/Output(s)

```
NOTE: 11 rows were updated in LIB78400.TESTING_DS.
```

10.1.6.3 Description

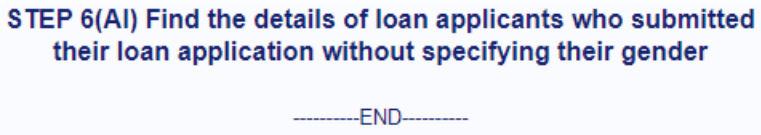
The data scientist would like to impute the 11 missing values found in GENDER with its mode, male. As a result, the 11 rows with missing values for GENDER were updated with male gender in LIB78400.TESTING_DS.

10.1.7 Step 6: List Out the Loan Applicants Who Did Not Detail Their Gender During the Loan Application Submission (After Imputation)

10.1.7.1 SAS Codes

```
/* STEP 6: (AI) Find the details of loan applicants who submitted their loan application without gender */
TITLE1 'STEP 6(AI) Find the details of loan applicants who submitted';
TITLE2 'their loan application without specifying their gender';
FOOTNOTE '----------END----------';

PROC SQL;

SELECT *
FROM LIB78400.TESTING_DS e
WHERE ( e.GENDER eq '' OR e.GENDER IS MISSING );

QUIT;
```

10.1.7.2 Screenshot(s)/Output(s)



10.1.7.3 Description

After imputation, the data scientist would like to double-check if the details of loan applicants who did not specify their gender were still in the testing set. As a result, the details could not be found as the missing values for GENDER were imputed with the mode ‘male’.

10.2 Imputing Missing Values Found in the Categorical Variable – EMPLOYMENT

10.2.1 Step 1: List Out the Loan Applicants Who Did Not Detail Their Employment Status During the Loan Application Submission

10.2.1.1 SAS Codes

```
1215 /* STEP 1: Find the details of loan applicants who submitted their
1216 loan application without specifying their employment status */
1217 TITLE1 'Find the details of loan applicants who submitted their';
1218 TITLE2 'loan application without specifying their employment status';
1219 FOOTNOTE '----------END----------';
1220
1221 PROC SQL;
1222
1223 SELECT *
1224 FROM LIB78400.TESTING_DS e
1225 WHERE ( e.EMPLOYMENT eq '' OR e.EMPLOYMENT IS MISSING );
1226
1227 QUIT;
```

10.2.1.2 Screenshot(s)/Output(s)

Find the details of loan applicants who submitted their loan application without specifying their employment status

SME_LOAN_ID_NO	GENDER	MARITAL_STATUS	FAMILY_MEMBERS	QUALIFICATION	EMPLOYMENT	CANDIDATE_INCOME	GUARANTEE_INCOME	LOAN_AMOUNT	LOAN_DURATION	LOAN_HISTORY	LOAN_LOCATION	LOAN_APPROVAL_STATUS
LP001059	Male	Married	2	Graduate		13633	0	280	240	1	City	
LP001082	Male	Married	1	Graduate		2185	1516	162	360	1	Town	
LP001094	Male	Married	2	Graduate		12173	0	166	360	0	Town	
LP001208	Male	Married	2	Graduate		7350	4029	185	180	1	City	
LP001375	Male	Married	1	Graduate		4083	1775	139	60	1	City	
LP001472	Female	Not Married	0	Graduate		5058	0	200	360	1	Village	
LP001789	Male	Married	3+	Under Graduate		6794	528	139	360	0	City	
LP001906	Male	Not Married	0	Graduate		2964	0	84	360	0	Town	
LP001950	Female	Married	3+	Graduate		1750	2935	94	360	0	Town	
LP001999	Male	Married	2	Graduate		4912	4614	160	360	1	Village	
LP002069	Male	Married	2	Under Graduate		3785	2912	180	360	0	Village	
LP002346	Male	Married	0	Graduate		2539	1704	125	360	0	Village	
LP002399	Male	Not Married	0	Graduate		2858	0	123	360	0	Village	
LP002415	Female	Not Married	1	Graduate		1850	4583	81	360	.	Village	
LP002542	Male	Married	0	Graduate		6500	0	144	360	1	City	
LP002551	Male	Married	3+	Under Graduate		3634	910	176	360	0	Town	
LP002572	Male	Married	1	Graduate		8750	0	297	360	1	City	
LP002584	Male	Not Married	0	Graduate		1972	4347	106	360	1	Village	
LP002610	Male	Married	1	Under Graduate		1792	2565	128	360	1	City	
LP002630	Male	Not Married	0	Under Graduate		3808	0	83	360	1	Village	
LP002651	Male	Married	1	Graduate		6300	0	125	360	0	City	
LP002791	Male	Not Married	1	Graduate		16000	5000	40	360	1	Town	
LP002803	Male	Married	1	Under Graduate		2600	618	122	360	1	Town	

----------END----------

10.2.1.3 Description

The data scientist would like to discover which loan applicants did not specify their employment status during the application submission stage. There were 20 male and three female applicants who did not specify their employment status. Their monthly income ranges between $1,792 and $16,000 and loan duration ranges between 60 and 360 months.

10.2.2 Step 2: Count the Loan Applicants Who Did Not Detail Their Employment Status During the Loan Application Submission

10.2.2.1 SAS Codes

```
1229 /* STEP 2: Count the number of loan applicants who submitted
1230 their loan application without specifying their employment status */
1231 TITLE1 'Count the number of loan applicants who submitted their loan';
1232 TITLE2 'application without specifying their employment status';
1233 FOOTNOTE '----------END----------';
1234
1235 PROC SQL;
1236
1237 SELECT COUNT(*) Label = 'Number of Applicants'
1238 FROM LIB78400.TESTING_DS e
1239 WHERE ( e.EMPLOYMENT eq '' OR e.EMPLOYMENT IS MISSING );
1240
1241 QUIT;
```

10.2.2.2 Screenshot(s)/Output(s)

Count the number of loan applicants who submitted their loan application without specifying their employment status

Number of Applicants
23

---------END---------

10.2.2.3 Description

The data scientist would like to calculate the number of loan applicants who did not specify their employment status during the application submission stage. As a result, 23 applicants did not specify their employment status.

10.2.3 Step 3: Find the Statistics of Employed and Unemployed Loan Applicants

10.2.3.1 SAS Codes

```
1243 /* STEP 3: Find the statistics of employed and unemployed applicants */
1244 TITLE1 'Find the statistics of employed & unemployed applicants';
1245
1246 PROC SQL;
1247
1248 SELECT e.EMPLOYMENT AS EMPLOYMENT,
1249        COUNT(*) AS COUNTS
1250 FROM LIB78400.TESTING_DS e
1251 WHERE ( e.EMPLOYMENT ne '' OR e.EMPLOYMENT IS NOT MISSING )
1252 GROUP BY e.EMPLOYMENT;
1253
1254 QUIT;
```

10.2.3.2 Screenshot(s)/Output(s)

Find the statistics of employed & unemployed applicants

EMPLOYMENT	COUNTS
No	307
Yes	37

10.2.3.3 Description

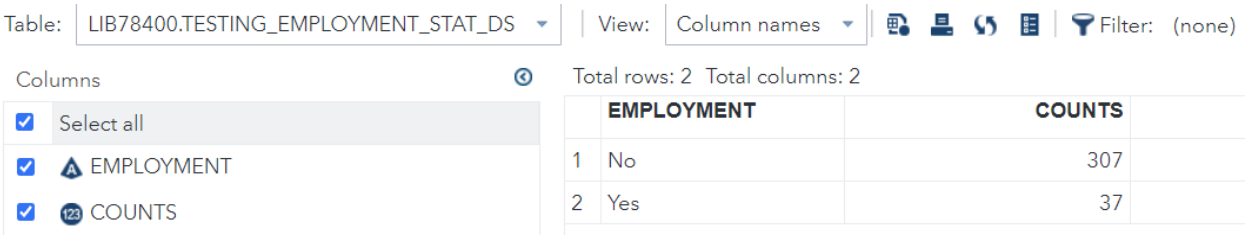
The data scientist would like to discover the statistics of employed and non-employed applicants. As a result, there were 3307 unemployed applicants and 37 employed applicants.

10.2.4 Step 4: Save the Statistics of Employed and Unemployed Loan Applicants in a Dataset

10.2.4.1 SAS Codes

```
1256 /* STEP 4: Save the statistics in a dataset */
1257
1258 PROC SQL;
1259
1260 CREATE TABLE LIB78400.TESTING_EMPLOYMENT_STAT_DS AS
1261 SELECT e.EMPLOYMENT AS EMPLOYMENT,
1262        COUNT(*) AS COUNTS
1263 FROM LIB78400.TESTING_DS e
1264 WHERE ( e.EMPLOYMENT ne '' OR e.EMPLOYMENT IS NOT MISSING )
1265 GROUP BY e.EMPLOYMENT;
1266
1267 QUIT;
```

10.2.4.2 Screenshot(s)/Output(s)



	EMPLOYMENT	COUNTS
1	No	307
2	Yes	37

10.2.4.3 Description

The data scientist would like to save the statistics of employed and unemployed applicants into a new dataset that would be stored in the LIB78400 library. As a result, a new dataset called LIB78400.TESTING_EMPLOYMENT_STAT_DS was created.

10.2.5 Step 4.1: Make a Backup Copy of the Dataset Created

10.2.5.1 SAS Codes

```
1269 /* STEP 4.1: Make a backup copy of the dataset */
1270
1271 PROC SQL;
1272
1273 CREATE TABLE LIB78400.TESTING_BK_DS AS
1274 SELECT *
1275 FROM LIB78400.TESTING_DS;
1276
1277 QUIT;
```

10.2.5.2 Screenshot(s)/Output(s)

Table: LIB78400.TESTING_BK_DS | View: Column names | Filter: (none)

Columns

- Select all
- SME_LOAN_ID_NO
- GENDER
- MARITAL_STATUS
- FAMILY_MEMBERS
- QUALIFICATION
- EMPLOYMENT
- CANDIDATE_INCOME
- GUARANTEE_INCOME
- LOAN_AMOUNT
- LOAN_DURATION

Property	Value
Label	
Name	
Length	
Type	
Format	

Total rows: 367 Total columns: 13 — Rows 1-100

	SME_LOAN_ID...	GEND...	MARITAL_STA...	FAMILY_MEMB...	QUALIFICATION	EMPLOYM...	CANDIDATE_INCOME	GUARANTEE_INCOME
1	LP001015	Male	Married	0	Graduate	No	5720	0
2	LP001022	Male	Married	1	Graduate	No	3076	1500
3	LP001031	Male	Married	2	Graduate	No	5000	1800
4	LP001035	Male	Married	2	Graduate	No	2340	2546
5	LP001051	Male	Not Married	0	Under Graduate	No	3276	0
6	LP001054	Male	Married	0	Under Graduate	Yes	2165	3422
7	LP001055	Female	Not Married	1	Under Graduate	No	2226	0
8	LP001056	Male	Married	2	Under Graduate	No	3881	0
9	LP001059	Male	Married	2	Graduate		13633	0
10	LP001067	Male	Not Married	0	Under Graduate	No	2400	2400
11	LP001078	Male	Not Married	0	Under Graduate	No	3091	0
12	LP001082	Male	Married	1	Graduate		2185	1516
13	LP001083	Male	Not Married	3+	Graduate	No	4166	0
14	LP001094	Male	Married	2	Graduate		12173	0
15	LP001096	Female	Not Married	0	Graduate	No	4666	0
16	LP001099	Male	Not Married	1	Graduate	No	5667	0
17	LP001105	Male	Married	2	Graduate	No	4583	2916

10.2.5.3 Description

The data scientist would like to make a backup copy of the testing set so that he can access it whenever the original testing set is corrupted during analysis. As a result, a backup copy called LIB78400.TESTING_BK_DS was created.

10.2.6 Step 5: Impute Missing Values in the Categorical Variable – EMPLOYMENT

10.2.6.1 SAS Codes

```
1279 /* STEP 5: Impute the missing values found in the categorical variable - EMPLOYMENT */
1280
1281 PROC SQL;
1282
1283 UPDATE LIB78400.TESTING_DS
1284 SET EMPLOYMENT = ( SELECT to.EMPLOYMENT AS EMPLOYMENT
1285                FROM LIB78400.TESTING_EMPLOYMENT_STAT_DS to
1286                WHERE to.COUNTS eq ( SELECT MAX(ti.COUNTS) AS HIGHEST_COUNT
1287                                     FROM LIB78400.TESTING_EMPLOYMENT_STAT_DS ti ) )
1288                                     /* Above is a sub-program to find the highest count */
1289 WHERE ( EMPLOYMENT eq '' OR EMPLOYMENT IS MISSING );
1290
1291 QUIT;
```

10.2.6.2 Screenshot(s)/Output(s)

```
NOTE: 23 rows were updated in LIB78400.TESTING_DS.
```

10.2.6.3 Description

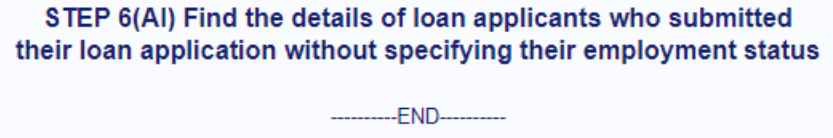
The data scientist would like to impute the 23 missing values found in EMPLOYMENT with its mode, ‘No’. As a result, the 23 rows with missing values for EMPLOYMENT were updated with ‘No’ in LIB78400.TESTING_DS.

10.2.7 Step 6: List Out the Loan Applicants Who Did Not Detail Their Employment Status During the Loan Application Submission (After Imputation)

10.2.7.1 SAS Codes

```
/* STEP 6: (AI) Find the details of loan applicants who submitted their loan application without employment status */
TITLE1 'STEP 6(AI) Find the details of loan applicants who submitted';
TITLE2 'their loan application without specifying their employment status';
FOOTNOTE '----------END----------';

PROC SQL;

SELECT *
FROM LIB78400.TESTING_DS e
WHERE ( e.EMPLOYMENT eq '' OR e.EMPLOYMENT IS MISSING );

QUIT;
```

10.2.7.2 Screenshot(s)/Output(s)



10.2.7.3 Description

After imputation, the data scientist would like to double-check if the details of loan applicants who did not specify their employment status were still in the testing set. As a result, the details could not be found as the missing values for EMPLOYMENT were imputed with the mode ‘No’.

10.3 Imputing Missing Values Found in the Categorical Variable – LOAN_HISTORY

10.3.1 Step 1: List Out the Loan Applicants Without Loan History During the Loan Application Submission

10.3.1.1 SAS Codes

```
1445 /* STEP 1: Find the details of loan applicants who submitted their
1446 loan application without loan history */
1447 TITLE1 'Find the details of loan applicants who submitted their';
1448 TITLE2 'loan application without loan history';
1449 FOOTNOTE '----------END----------';
1450
1451 PROC SQL;
1452
1453 SELECT *
1454 FROM LIB78400.TESTING_DS e
1455 WHERE ( e.LOAN_HISTORY eq . OR e.LOAN_HISTORY IS MISSING );
1456
1457 QUIT;
```

10.3.1.2 Screenshot(s)/Output(s)

Find the details of loan applicants who submitted their
loan application without loan history

SME_LOAN_ID_NO	GENDER	MARITAL_STATUS	FAMILY_MEMBERS	QUALIFICATION	EMPLOYMENT	CANDIDATE_INCOME	GUARANTEE_INCOME	LOAN_AMOUNT	LOAN_DURATION	LOAN_HISTORY	LOAN_LOCATION	LOAN_APPROVAL_STATUS
LP001035	Male	Married	2	Graduate	No	2340	2546	100	360	.	City	
LP001083	Male	Not Married	3	Graduate	No	4166	0	40	180	.	City	
LP001163	Male	Married	2	Graduate	No	4363	1250	140	360	.	City	
LP001174	Male	Married	0	Graduate	No	3772	833	57	360	.	Town	
LP001232	Male	Married	0	Graduate	No	4260	3900	185	.	.	City	
LP001475	Male	Married	0	Graduate	Yes	3188	2286	130	360	.	Village	
LP001527	Male	Married	3	Graduate	No	6835	0	188	360	.	Town	
LP001558	Male	Not Married	0	Graduate	No	2243	2233	107	360	.	Town	
LP001601	Male	Not Married	3	Graduate	No	4243	4123	157	360	.	Town	
LP001771	Female	Not Married	3	Graduate	No	4083	0	103	360	.	Town	
LP001791	Male	Married	0	Graduate	Yes	32000	0	550	360	.	Town	
LP001921	Male	Not Married	1	Graduate	No	3180	2370	80	240	.	Village	
LP002009	Female	Not Married	0	Graduate	No	2918	0	65	360	.	Village	
LP002017	Male	Married	3	Graduate	No	15312	0	187	360	.	City	
LP002046	Male	Married	0	Under Graduate	No	4483	0	135	360	.	Town	
LP002111	Male	Married	0	Graduate	No	3016	1300	100	360	.	City	
LP002212	Male	Married	0	Graduate	No	2166	2166	108	360	.	City	
LP002415	Female	Not Married	1	Graduate	No	1850	4583	81	360	.	Village	
LP002425	Male	Not Married	0	Graduate	No	3417	738	100	360	.	Village	
LP002441	Male	Not Married	0	Graduate	No	3579	3308	138	360	.	Town	
LP002566	Female	Not Married	0	Graduate	No	5530	0	135	360	.	City	
LP002581	Male	Married	0	Under Graduate	No	2157	2730	140	360	.	Village	
LP002712	Male	Not Married	2	Under Graduate	No	2733	1083	180	360	.	Town	
LP002816	Male	Married	1	Graduate	No	3500	1658	104	360	.	Town	
LP002853	Female	Not Married	0	Under Graduate	No	3015	2000	145	360	.	City	
LP002901	Male	Not Married	0	Graduate	No	2283	15000	106	360	.	Village	
LP002954	Male	Married	2	Under Graduate	No	3132	0	76	360	.	Village	
LP002965	Female	Married	0	Graduate	No	8550	4255	96	360	.	City	
LP002980	Male	Not Married	0	Graduate	No	3250	1993	126	360	.	Town	

----------END----------

10.3.1.3 Description

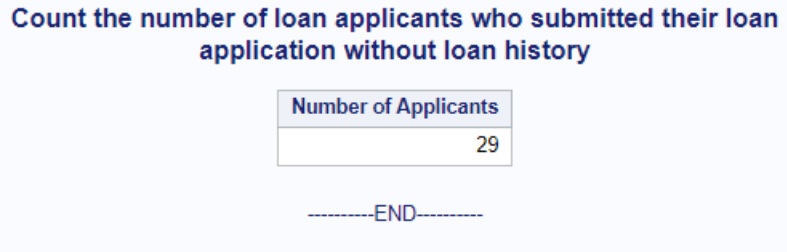
The data scientist would like to discover which loan applicants had the missing loan history during the application submission stage. There were 23 male and six female applicants who had missing loan histories. Their monthly income ranges between $1,850 and 32,000 and loan duration ranges between 180 and 360 months.

10.3.2 Step 2: Count the Loan Applicants Without Loan History During the Loan Application Submission

10.3.2.1 SAS Codes

```
1459 /* STEP 2: Count the number of loan applicants who submitted
1460 their loan application without loan history */
1461 TITLE1 'Count the number of loan applicants who submitted their loan';
1462 TITLE2 'application without loan history';
1463 FOOTNOTE '----------END----------';
1464
1465 PROC SQL;
1466
1467 SELECT COUNT(*) Label = 'Number of Applicants'
1468 FROM LIB78400.TESTING_DS e
1469 WHERE ( e.LOAN_HISTORY eq . OR e.LOAN_HISTORY IS MISSING );
1470
1471 QUIT;
```

10.3.2.2 Screenshot(s)/Output(s)



Count the number of loan applicants who submitted their loan application without loan history

Number of Applicants
29

----------END----------

10.3.2.3 Description

The data scientist would like to calculate the number of loan applicants who had missing loan histories during the application submission stage. As a result, 29 applicants did not specify their loan history.

10.3.3 Step 3: Find the Statistics of Loan Applicants with Positive and Negative Loan History

10.3.3.1 SAS Codes

```
1473 /* STEP 3: Find the statistics of applicants with positive and negative past loan records */
1474 TITLE1 'Find the statistics of applicants with positive and negative past loan records';
1475
1476 PROC SQL;
1477
1478 SELECT e.LOAN_HISTORY AS LOAN_HISTORY,
1479        COUNT(*) AS COUNTS
1480 FROM LIB78400.TESTING_DS e
1481 WHERE ( e.LOAN_HISTORY ne . OR e.LOAN_HISTORY IS NOT MISSING )
1482 GROUP BY e.LOAN_HISTORY;
1483
1484 QUIT;
```

10.3.3.2 Screenshot(s)/Output(s)

Find the statistics of applicants with positive and negative past loan records

LOAN_HISTORY	COUNTS
0	59
1	279

10.3.3.3 Description

The data scientist would like to discover the statistics of loan applicants with positive and negative loan records. As a result, there were 59 applicants with negative loan history and 279 applicants with positive loan history.

10.3.4 Step 4: Save the Statistics of Loan Applicants with Positive and Negative Loan History in a Dataset

10.3.4.1 SAS Codes

```
1486 /* STEP 4: Save the statistics in a dataset */
1487
1488 PROC SQL;
1489
1490 CREATE TABLE LIB78400.TESTING_LH_STAT_DS AS
1491 SELECT e.LOAN_HISTORY AS LOAN_HISTORY,
1492        COUNT(*) AS COUNTS
1493 FROM LIB78400.TESTING_DS e
1494 WHERE ( e.LOAN_HISTORY ne . OR e.LOAN_HISTORY IS NOT MISSING )
1495 GROUP BY e.LOAN_HISTORY;
1496
1497 QUIT;
```

10.3.4.2 Screenshot(s)/Output(s)

Table: LIB78400.TESTING_LH_STAT_DS | View: Column names | Filter: (none)

Columns

- Select all
- LOAN_HISTORY
- COUNTS

Total rows: 2 Total columns: 2

	LOAN_HISTORY	COUNTS
1	0	59
2	1	279

10.3.4.3 Description

The data scientist would like to save the statistics of loan applicants with positive and negative past loan records into a new dataset that would be stored in the LIB78400 library. As a result, a new dataset called LIB78400.TESTING_LH_STAT_DS was created.

10.3.5 Step 4.1: Make a Backup Copy of the Dataset Created

10.3.5.1 SAS Codes

```
1499 /* STEP 4.1: Make a backup copy of the dataset */
1500
1501 PROC SQL;
1502
1503 CREATE TABLE LIB78400.TESTING_BK_DS AS
1504 SELECT *
1505 FROM LIB78400.TESTING_DS;
1506
1507 QUIT;
```

10.3.5.2 Screenshot(s)/Output(s)

Table: LIB78400.TESTING_BK_DS | View: Column names | Filter: (none)

Columns

- Select all
- SME_LOAN_ID_NO
- GENDER
- MARITAL_STATUS
- FAMILY_MEMBERS
- QUALIFICATION
- EMPLOYMENT
- CANDIDATE_INCOME
- GUARANTEE_INCOME
- LOAN_AMOUNT
- LOAN_DURATION

Property	Value
Label	
Name	
Length	
Type	
Format	

Total rows: 367 Total columns: 13 Rows 1-100

	SME_LOAN_ID...	GEND...	MARITAL_STA...	FAMILY_MEMB...	QUALIFICATION	EMPLOYM...	CANDIDATE_INCOME	GUARANTEE_INCOME
1	LP001015	Male	Married	0	Graduate	No	5720	0
2	LP001022	Male	Married	1	Graduate	No	3076	1500
3	LP001031	Male	Married	2	Graduate	No	5000	1800
4	LP001035	Male	Married	2	Graduate	No	2340	2546
5	LP001051	Male	Not Married	0	Under Graduate	No	3276	0
6	LP001054	Male	Married	0	Under Graduate	Yes	2165	3422
7	LP001055	Female	Not Married	1	Under Graduate	No	2226	0
8	LP001056	Male	Married	2	Under Graduate	No	3881	0
9	LP001059	Male	Married	2	Graduate	No	13633	0
10	LP001067	Male	Not Married	0	Under Graduate	No	2400	2400
11	LP001078	Male	Not Married	0	Under Graduate	No	3091	0
12	LP001082	Male	Married	1	Graduate	No	2185	1516
13	LP001083	Male	Not Married	3	Graduate	No	4166	0
14	LP001094	Male	Married	2	Graduate	No	12173	0
15	LP001096	Female	Not Married	0	Graduate	No	4666	0
16	LP001099	Male	Not Married	1	Graduate	No	5667	0
17	LP001105	Male	Married	2	Graduate	No	4583	2916

10.3.5.3 Description

The data scientist would like to make a backup copy of the testing set so that he can access it whenever the original testing set is corrupted during analysis. As a result, a backup copy called LIB78400.TESTING_BK_DS was created.

10.3.6 Step 5: Impute Missing Values in the Categorical Variable – LOAN_HISTORY

10.3.6.1 SAS Codes

```
/* STEP 5: Impute the missing values found in the categorical variable - LOAN_HISTORY */

PROC SQL;

UPDATE LIB78400.TESTING_DS
SET LOAN_HISTORY = ( SELECT to.LOAN_HISTORY AS LOAN_HISTORY
                     FROM LIB78400.TESTING_LH_STAT_DS to
                     WHERE to.COUNTS eq ( SELECT MAX(ti.COUNTS) AS HIGHEST_COUNT
                                          FROM LIB78400.TESTING_LH_STAT_DS ti ) )
                                        /* Above is a sub-program to find the highest count */
WHERE ( LOAN_HISTORY eq . OR LOAN_HISTORY IS MISSING );

QUIT;
```

10.3.6.2 Screenshot(s)/Output(s)

```
NOTE: 29 rows were updated in LIB78400.TESTING_DS.
```

10.3.6.3 Description

The data scientist would like to impute the 29 missing values found in LOAN_HISTORY with its mode, ‘1’ indicating positive loan records. As a result, the 29 rows with missing values for LOAN_HISTORY were updated with the binary value ‘1’ in LIB78400.TESTING_DS.

10.3.7 Step 6: List Out the Loan Applicants Without Loan History During the Loan Application Submission (After Imputation)

10.3.7.1 SAS Codes

```
/* STEP 6: (AI) Find the details of loan applicants who submitted their loan application without employment status */
TITLE1 'STEP 6(AI) Find the details of loan applicants who submitted';
TITLE2 'their loan application without loan history';
FOOTNOTE '----------END----------';

PROC SQL;

SELECT *
FROM LIB78400.TESTING_DS e
WHERE ( e.LOAN_HISTORY eq . OR e.LOAN_HISTORY IS MISSING );

QUIT;
```

10.3.7.2 Screenshot(s)/Output(s)

STEP 6(AI) Find the details of loan applicants who submitted their loan application without loan history

----------END----------

10.3.7.3 Description

After imputation, the data scientist would like to double-check if the details of loan applicants without loan history were still in the testing set. As a result, the details could not be found as the missing values for LOAN_HISTORY were imputed with the binary-valued mode, '1'.

11.0 Imputing Missing Values Found in the Continuous Variables of LIB78400.TESTING_DS

11.1 Imputing Missing Values Found in the Continuous Variable – MARITAL_STATUS

11.1.1 Step 1: List Out the Loan Applicants Who Did Not Detail Their Loan Amount During the Loan Application Submission

11.1.1.1 SAS Codes

```
1538 /* STEP 1: Find the details of loan applicants who submitted their
1539 loan application without specifying loan amount */
1540 TITLE1 'Find the details of loan applicants who submitted their';
1541 TITLE2 'loan application without specifying loan amount';
1542 FOOTNOTE '----------END----------';
1543
1544 PROC SQL;
1545
1546 SELECT *
1547 FROM LIB78400.TESTING_DS e
1548 WHERE ( e.LOAN_AMOUNT eq . OR e.LOAN_AMOUNT IS MISSING );
1549
1550 QUIT;
```

11.1.1.2 Screenshot(s)/Output(s)

Find the details of loan applicants who submitted their loan application without specifying loan amount

SME_LOAN_ID_NO	GENDER	MARITAL_STATUS	FAMILY_MEMBERS	QUALIFICATION	EMPLOYMENT	CANDIDATE_INCOME	GUARANTEE_INCOME	LOAN_AMOUNT	LOAN_DURATION	LOAN_HISTORY	LOAN_LOCATION	LOAN_APPROVAL_STATUS
LP001415	Male	Married	1	Graduate	No	3413	4053	.	360	1	Town	
LP001542	Female	Married	0	Graduate	No	2262	0	.	480	0	Town	
LP002057	Male	Married	0	Under Graduate	No	13083	0	.	360	1	Village	
LP002360	Male	Married	0	Graduate	No	10000	0	.	360	1	City	
LP002593	Male	Married	1	Graduate	No	8333	4000	.	360	1	City	

----------END----------

11.1.1.3 Description

The data scientist would like to discover which loan applicants did not specify their applied loan amount during the application submission stage. There were four male and one female unemployed applicants who did not specify their loan amount. Their monthly income ranges between $2,262 and $13,083 and loan duration ranges between 360 and 480 months.

11.1.2 Step 2: Count the Loan Applicants Who Did Not Detail Their Loan Amount During the Loan Application Submission

11.1.2.1 SAS Codes

```
1552 /* STEP 2: Count the number of loan applicants who submitted
1553 their loan application without specifying loan amount */
1554 TITLE1 'Count the number of loan applicants who submitted their loan';
1555 TITLE2 'application without specifying loan amount';
1556 FOOTNOTE '----------END----------';
1557
1558 PROC SQL;
1559
1560 SELECT COUNT(*) Label = 'Number of Applicants'
1561 FROM LIB78400.TESTING_DS e
1562 WHERE ( e.LOAN_AMOUNT eq . OR e.LOAN_AMOUNT IS MISSING );
1563
1564 QUIT;
```

11.1.2.2 Screenshot(s)/Output(s)

Count the number of loan applicants who submitted their loan application without specifying loan amount

Number of Applicants
5

----------END----------

11.1.2.3 Description

The data scientist would like to calculate the number of loan applicants who did not specify their loan amount during the application submission stage. As a result, five applicants did not specify their loan amount.

11.1.3 Step 3: Impute Missing Values in the Continuous Variable – LOAN_AMOUNT

11.1.3.1 SAS Codes

```
/* STEP 3: Impute the missing values found in the continuous variable - LOAN_AMOUNT */

PROC STDIZE DATA = LIB78400.TESTING_DS REPONLY

METHOD = MEAN OUT = LIB78400.TESTING_DS;
VAR LOAN_AMOUNT;

QUIT;
```

11.1.3.2 Screenshot(s)/Output(s)

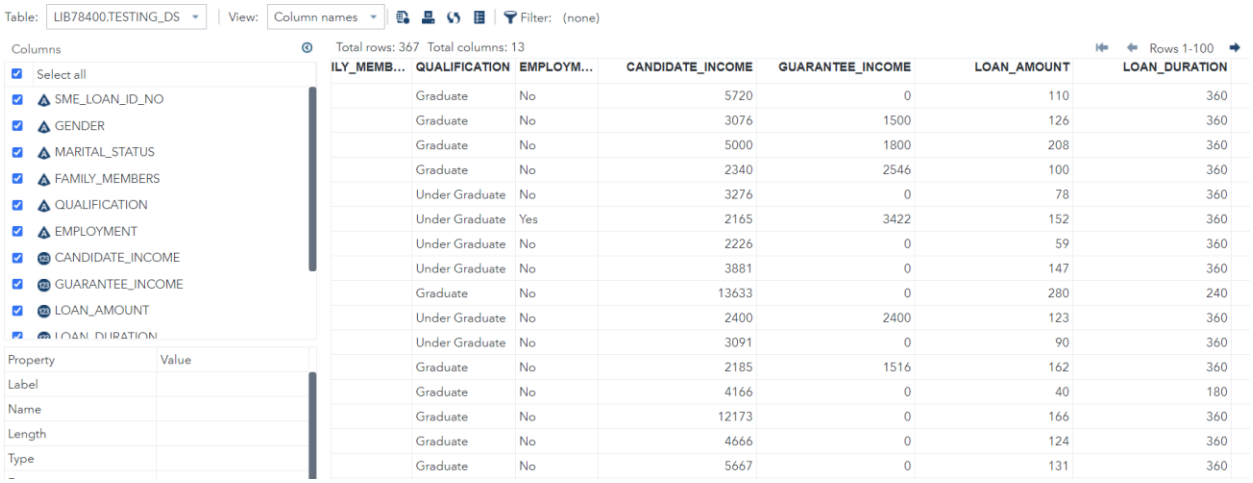


Table: LIB78400.TESTING_DS | View: Column names | Filter: (none)

Columns: Select all, SME_LOAN_ID_NO, GENDER, MARITAL_STATUS, FAMILY_MEMBERS, QUALIFICATION, EMPLOYMENT, CANDIDATE_INCOME, GUARANTEE_INCOME, LOAN_AMOUNT, LOAN_DURATION

Property | Value: Label, Name, Length, Type

Total rows: 367 Total columns: 13 — Rows 1-100

ILY_MEMB...	QUALIFICATION	EMPLOYM...	CANDIDATE_INCOME	GUARANTEE_INCOME	LOAN_AMOUNT	LOAN_DURATION
	Graduate	No	5720	0	110	360
	Graduate	No	3076	1500	126	360
	Graduate	No	5000	1800	208	360
	Graduate	No	2340	2546	100	360
	Under Graduate	No	3276	0	78	360
	Under Graduate	Yes	2165	3422	152	360
	Under Graduate	No	2226	0	59	360
	Under Graduate	No	3881	0	147	360
	Graduate	No	13633	0	280	240
	Under Graduate	No	2400	2400	123	360
	Under Graduate	No	3091	0	90	360
	Graduate	No	2185	1516	162	360
	Graduate	No	4166	0	40	180
	Graduate	No	12173	0	166	360
	Graduate	No	4666	0	124	360
	Graduate	No	5667	0	131	360

11.1.3.3 Description

The data scientist would like to impute the five missing values found in LOAN_AMOUNT with the average value of LOAN_AMOUNT. As a result, the five rows with missing values for LOAN_AMOUNT were updated with the mean value of LOAN_AMOUNT in LIB78400.TESTING_DS.

11.1.4 Step 4 & 5: List Out and Count the Loan Applicants Who Did Not Detail Their Loan Amount During the Loan Application Submission (After Imputation)

11.1.4.1 SAS Codes

```
1585 /* STEP 4: (AI) Find the details of loan applicants who submitted their
1586 loan application without specifying loan amount */
1587 TITLE1 'Find the details of loan applicants who submitted their';
1588 TITLE2 'loan application without specifying loan amount';
1589 FOOTNOTE '----------END----------';
1590
1591 PROC SQL;
1592
1593 SELECT *
1594 FROM LIB78400.TESTING_DS e
1595 WHERE ( e.LOAN_AMOUNT eq . OR e.LOAN_AMOUNT IS MISSING );
1596
1597 QUIT;
1598
1599 /* STEP 5: (AI) Count the number of loan applicants who submitted
1600 their loan application without specifying loan amount */
1601 TITLE1 'Count the number of loan applicants who submitted their loan';
1602 TITLE2 'application without specifying loan amount';
1603 FOOTNOTE '----------END----------';
1604
1605 PROC SQL;
1606
1607 SELECT COUNT(*) Label = 'Number of Applicants'
1608 FROM LIB78400.TESTING_DS e
1609 WHERE ( e.LOAN_AMOUNT eq . OR e.LOAN_AMOUNT IS MISSING );
1610
1611 QUIT;
```

11.1.4.2 Screenshot(s)/Output(s)

Find the details of loan applicants who submitted their loan application without specifying loan amount

----------END----------

Count the number of loan applicants who submitted their loan application without specifying loan amount

Number of Applicants
0

----------END----------

11.1.4.3 Description

After imputation, the data scientist would like to double-check if the details of loan applicants who did not specify their loan amount were still in the testing set. As a result, the details and number of applicants could not be found as the missing values for LOAN_AMOUNT were imputed with the mean value of LOAN_AMOUNT.

11.2 Imputing Missing Values Found in the Continuous Variable – LOAN_DURATION

11.2.1 Step 1: List Out the Loan Applicants Who Did Not Detail Loan Duration During the Loan Application Submission

11.2.1.1 SAS Codes

```
1615 /* STEP 1: Find the details of loan applicants who submitted their
1616 loan application without specifying loan duration */
1617 TITLE1 'Find the details of loan applicants who submitted their';
1618 TITLE2 'loan application without specifying loan duration';
1619 FOOTNOTE '----------END----------';
1620
1621 PROC SQL;
1622
1623 SELECT *
1624 FROM LIB78400.TESTING_DS e
1625 WHERE ( e.LOAN_DURATION eq . OR e.LOAN_DURATION IS MISSING );
1626
1627 QUIT;
```

11.2.1.2 Screenshot(s)/Output(s)

Find the details of loan applicants who submitted their
loan application without specifying loan duration

SME_LOAN_ID_NO	GENDER	MARITAL_STATUS	FAMILY_MEMBERS	QUALIFICATION	EMPLOYMENT	CANDIDATE_INCOME	GUARANTEE_INCOME	LOAN_AMOUNT	LOAN_DURATION	LOAN_HISTORY	LOAN_LOCATION	LOAN_APPROVAL_STATUS
LP001232	Male	Married	0	Graduate	No	4260	3900	185	.	1	City	
LP001268	Male	Not Married	0	Graduate	No	6792	3338	187	.	1	City	
LP001611	Male	Married	1	Graduate	No	1516	2900	80	.	0	Village	
LP001695	Male	Married	1	Under Graduate	No	3321	2088	70	.	1	Town	
LP002045	Male	Married	3	Graduate	No	10166	750	150	.	1	City	
LP002183	Male	Married	0	Under Graduate	No	3754	3719	118	.	1	Village	

----------END----------

11.2.1.3 Description

The data scientist would like to discover which loan applicants did not specify their loan duration during the application submission stage. Six male, unemployed applicants did not specify their loan duration. Their monthly income ranges between $1,516 and $10,166 and the applied loan amount ranges between $70,000 and $187,000.

11.2.2 Step 2: Count the Loan Applicants Who Did Not Detail Loan Duration During the Loan Application Submission

11.2.2.1 SAS Codes

```
1629 /* STEP 2: Count the number of loan applicants who submitted
1630 their loan application without specifying loan duration */
1631 TITLE1 'Count the number of loan applicants who submitted their loan';
1632 TITLE2 'application without specifying loan duration';
1633 FOOTNOTE '----------END----------';
1634
1635 PROC SQL;
1636
1637 SELECT COUNT(*) Label = 'Number of Applicants'
1638 FROM LIB78400.TESTING_DS e
1639 WHERE ( e.LOAN_DURATION eq . OR e.LOAN_DURATION IS MISSING );
1640
1641 QUIT;
```

11.2.2.2 Screenshot(s)/Output(s)

Count the number of loan applicants who submitted their loan application without specifying loan duration

Number of Applicants
6

----------END----------

11.2.2.3 Description

The data scientist would like to calculate the number of loan applicants who did not specify their loan duration during the application submission stage. As a result, six applicants did not specify their loan duration.

11.2.3 Step 3: Impute Missing Values in the Continuous Variable – LOAN_DURATION

11.2.3.1 SAS Codes

```
/* STEP 3: Impute the missing values found in the continuous variable - LOAN_DURATION */

PROC STDIZE DATA = LIB78400.TESTING_DS REPONLY

METHOD = MEAN OUT = LIB78400.TESTING_DS;
VAR LOAN_DURATION;

QUIT;
```

11.2.3.2 Screenshot(s)/Output(s)

Table: LIB78400.TESTING_DS | View: Column names | Filter: (none)

Columns

- Select all
- SME_LOAN_ID_NO
- GENDER
- MARITAL_STATUS
- FAMILY_MEMBERS
- QUALIFICATION
- EMPLOYMENT
- CANDIDATE_INCOME
- GUARANTEE_INCOME
- LOAN_AMOUNT
- LOAN_DURATION

Property	Value
Label	
Name	
Length	
Type	
Format	

Total rows: 367 Total columns: 13 — Rows 1-100

NCOME	GUARANTEE_INCOME	LOAN_AMOUNT	LOAN_DURATION	LOAN_HISTORY	LOAN_LOCAT...	LOAN_APPROVAL_ST...
5720	0	110	360	1	City	
3076	1500	126	360	1	City	
5000	1800	208	360	1	City	
2340	2546	100	360	1	City	
3276	0	78	360	1	City	
2165	3422	152	360	1	City	
2226	0	59	360	1	Town	
3881	0	147	360	0	Village	
13633	0	280	240	1	City	
2400	2400	123	360	1	Town	
3091	0	90	360	1	City	
2185	1516	162	360	1	Town	
4166	0	40	180	1	City	
12173	0	166	360	0	Town	
4666	0	124	360	1	Town	
5667	0	131	360	1	City	
4583	2916	200	360	1	City	

11.2.3.3 Description

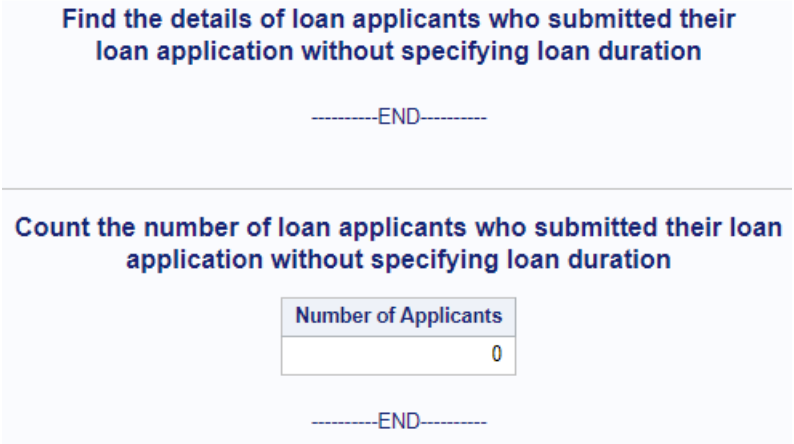
The data scientist would like to impute the six missing values found in LOAN_DURATION with the average value of LOAN_DURATION. As a result, the six rows with missing values for LOAN_DURATION were updated with the average value of LOAN_DURATION in LIB78400.TESTING_DS.

11.2.4 Step 4 & 5: List Out and Count the Loan Applicants Who Did Not Detail Their Loan Duration During the Loan Application Submission (After Imputation)

11.2.4.1 SAS Codes

```
1662 /* STEP 4: (AI) Find the details of loan applicants who submitted their
1663 loan application without specifying loan duration */
1664 TITLE1 'Find the details of loan applicants who submitted their';
1665 TITLE2 'loan application without specifying loan duration';
1666 FOOTNOTE '----------END----------';
1667
1668 PROC SQL;
1669
1670 SELECT *
1671 FROM LIB78400.TESTING_DS e
1672 WHERE ( e.LOAN_DURATION eq . OR e.LOAN_DURATION IS MISSING );
1673
1674 QUIT;
1675
1676 /* STEP 5: (AI) Count the number of loan applicants who submitted
1677 their loan application without specifying loan duration */
1678 TITLE1 'Count the number of loan applicants who submitted their loan';
1679 TITLE2 'application without specifying loan duration';
1680 FOOTNOTE '----------END----------';
1681
1682 PROC SQL;
1683
1684 SELECT COUNT(*) Label = 'Number of Applicants'
1685 FROM LIB78400.TESTING_DS e
1686 WHERE ( e.LOAN_DURATION eq . OR e.LOAN_DURATION IS MISSING );
1687
1688 QUIT;
```

11.2.4.2 Screenshot(s)/Output(s)



Find the details of loan applicants who submitted their loan application without specifying loan duration

----------END----------

Count the number of loan applicants who submitted their loan application without specifying loan duration

Number of Applicants
0

----------END----------

11.2.4.3 Description

After imputation, the data scientist would like to double-check if the details of the number of loan applicants who did not specify their loan duration were still in the testing set. As a result, the details could not be found as the missing values for LOAN_DURATION were imputed with the average value of LOAN_DURATION.

12.0 Model Development – Logistic Regression

12.1 SAS Codes

```
/* Model Development - Logistic Regression */

PROC LOGISTIC DATA = LIB78400.TRAINING_DS OUTMODEL = LIB78400.TRAINING_DS_LR_MODEL;
CLASS
    GENDER
    MARITAL_STATUS
    FAMILY_MEMBERS
    QUALIFICATION
    EMPLOYMENT
    LOAN_HISTORY
    LOAN_LOCATION
    ;

MODEL LOAN_APPROVAL_STATUS =
    GENDER
    MARITAL_STATUS
    FAMILY_MEMBERS
    QUALIFICATION
    EMPLOYMENT
    CANDIDATE_INCOME
    GUARANTEE_INCOME
    LOAN_AMOUNT
    LOAN_DURATION
    LOAN_HISTORY
    LOAN_LOCATION
    ;

OUTPUT OUT = LIB78400.TRAINING_OUT_DS P = PPRED_PROB;
/* PRED_PROB -> Predicted probability - variable to hold predicted probability
OUT -> the output will be stored in a dataset
Akaike Information Criterion (AIC) must be < SC (Schwarz Criterion) */
/* If Pr > ChiSq is <= 0.05, it means that the independent variable is an
important variable and as it is truly contributing to predicting the dependent variable */
RUN;
```

12.1.1 Number of Observations Read and Used

12.1.1.1 Screenshot(s)/Output(s)

Number of Observations Read	614
Number of Observations Used	614

Number of observations read & used are the same

12.1.1.2 Description

The number of observations read was similar to the number of observations used, which was 614. This was because of the imputation of missing values on categorical and continuous variables of both training and testing sets earlier.

12.1.2 Status of Model Convergence

12.1.2.1 Screenshot(s)/Output(s)

Model Convergence Status	
Convergence criterion (GCONV=1E-8)	satisfied

12.1.2.2 Description

The convergence criterion of the logistic regression model was satisfied. This indicated that the sigmoid curve with the best fit for the logistic regression model was found, with the parameter of the model achieving the highest possible accuracy.

12.1.3 Model Fit Statistics

12.1.3.1 Screenshot(s)/Output(s)

Model Fit Statistics		
Criterion	Intercept Only	Intercept and Covariates
AIC	764.891	587.154
SC	769.311	653.454
-2 Log L	762.891	557.154

12.1.3.2 Description

Based on the model fit statistics, the Schwarz Criterion (SC) value for intercept only, 769.311, was greater than the Akaike Information Criterion (AIC) value, 764.891. As the value of AIC must be lower than SC, the prerequisite was fulfilled.

12.1.4 Type 3 Analysis of Effects

12.1.4.1 Screenshot(s)/Output(s)

Type 3 Analysis of Effects

Effect	DF	Wald Chi-Square	Pr > ChiSq
GENDER	1	0.0100	0.9204
MARITAL_STATUS	1	5.3173	0.0211
FAMILY_MEMBERS	3	4.3866	0.2226
QUALIFICATION	1	2.4952	0.1142
EMPLOYMENT	1	0.0060	0.9384
CANDIDATE_INCOME	1	0.2268	0.6339
GUARANTEE_INCOME	1	2.2688	0.1320
LOAN_AMOUNT	1	1.4294	0.2319
LOAN_DURATION	1	0.5322	0.4657
LOAN_HISTORY	1	87.4798	<.0001
LOAN_LOCATION	2	12.0908	0.0024

12.1.4.2 Description

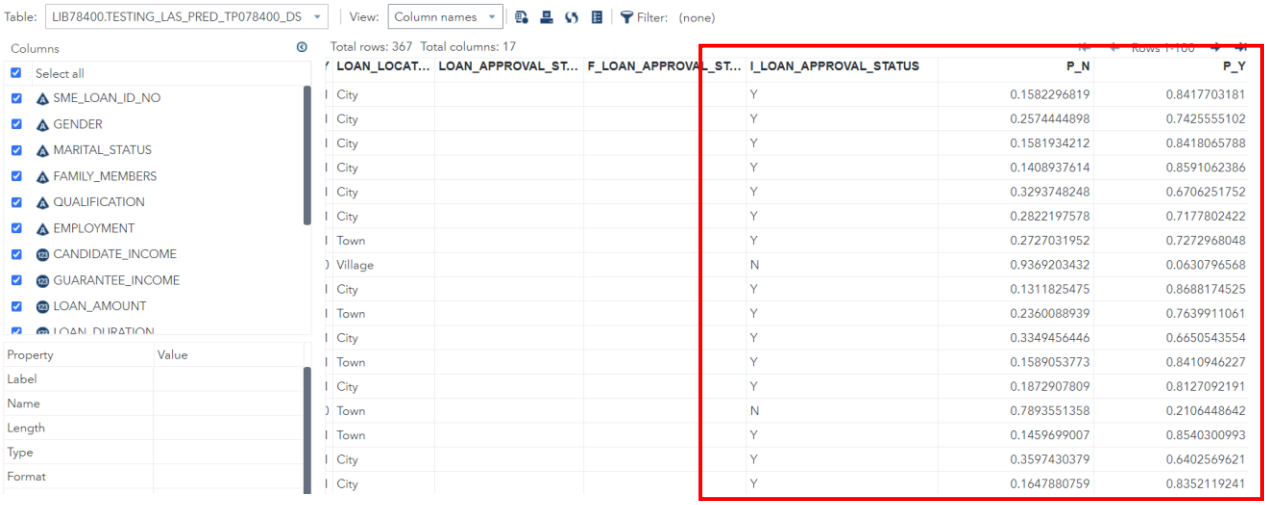
The p-value for the chi-square test must be less than or equal to 0.05 significance level so that the predictors are important variables in predicting the outcome variable. Therefore, marital status, qualification, loan history and loan location were important predictor variables in contributing towards predicting the outcome variable, which was loan approval status.

12.2 Forecasting Loan Approval Status Using the Previously Developed Logistic Regression Model

12.2.1 SAS Codes

```
1728 /*******************************************************
1729 Predict the loan approval status using the model created
1730 *******************************************************/
1731
1732 PROC LOGISTIC INMODEL = LIB78400.TRAINING_DS_LR_MODEL; /* Previously developed model */
1733
1734 SCORE DATA = LIB78400.TESTING_DS /* Test set */
1735 OUT = LIB78400.TESTING_LAS_PRED_TP078400_DS; /* Location where the output is stored */
1736
1737 QUIT;
```

12.2.2 Screenshot(s)/Output(s)



12.2.3 Description

The data scientist would like to use the previously developed logistic regression model to predict loan approval status on the testing set. The predicted outcomes were saved in a new dataset called LIB78400.TESTING_LAS_PRED_TP078400_DS. If the probability of loan approval was greater than the probability of loan rejection, the predicted outcome would be loan approval and vice versa.

12.3 Report Generation Using the SAS ODS – Output Delivery / Display System

12.3.1 SAS Codes

```
1742 TITLE 'List the status of the LOAN_APPROVAL_STATUS predicted';
1743 FOOTNOTE '----------End----------';
1744
1745 /* Generate and demonstrate the information of the loan approval status prediction */
1746
1747 PROC SQL;
1748
1749 SELECT *
1750 FROM LIB78400.TESTING_LAS_PRED_TP078400_DS;
1751
1752 QUIT;
```

```
1755 /* Generate the report using SAS ODS - Output Delivery / Display System
1756    Display the details of the loan approval status predicted */
1757
1758 ODS HTML CLOSE;
1759 ODS PDF CLOSE;
1760
1761 /* Determine the physical location of pdf */
1762 ODS PDF FILE = " /home/u63691887/DAP_FT_MAR_2024_TP078400/LAS_REPORT_TP078400.pdf";
1763 OPTIONS NODATE;
1764 TITLE1 'Prediction of Loan Approval Status';
1765 TITLE2 'APU,TPM';
1766 PROC REPORT DATA = LIB78400.TESTING_LAS_PRED_TP078400_DS NOWINDOWS;
1767 BY SME_LOAN_ID_NO;
1768 DEFINE SME_LOAN_ID_NO / GROUP 'SME LOAN ID';
1769 DEFINE GENDER / GROUP 'GENDER';
1770 DEFINE MARITAL_STATUS / GROUP 'MARITAL STATUS';
1771 DEFINE FAMILY_MEMBERS / GROUP 'NO. OF FAMILY MEMBERS';
1772 DEFINE CANDIDATE_INCOME / GROUP 'MONTHLY INCOME';
1773 DEFINE GUARANTEE_INCOME / GROUP 'CO-APPLICANT INCOME';
1774 DEFINE LOAN_AMOUNT / GROUP 'LOAN AMOUNT';
1775 DEFINE LOAN_DURATION / GROUP 'LOAN DURATION (MONTHS)';
1776 DEFINE LOAN_HISTORY / GROUP 'LOAN HISTORY';
1777 DEFINE LOAN_LOCATION / GROUP 'LOAN LOCATION';
1778 FOOTNOTE '----------End of Report----------';
1779 RUN;
```

12.3.2 Screenshot(s)/Output(s)

List the status of the LOAN_APPROVAL_STATUS predicted

...IFICATION	EMPLOYMENT	CANDIDATE_INCOME	GUARANTEE_INCOME	LOAN_AMOUNT	LOAN_DURATION	LOAN_HISTORY	LOAN_LOCATION	LOAN_APPROVAL_STATUS	From: LOAN_APPROVAL_STATUS	Into: LOAN_APPROVAL_STATUS	Predicted Probability: LOAN_APPROVAL_STATUS=N	Predicted Probability: LOAN_APPROVAL_STATUS=Y
...ate	No	5720	0	110	360	1	City			Y	0.15823	0.84177
...ate	No	3076	1500	126	360	1	City			Y	0.257444	0.742556
...ate	No	5000	1800	208	360	1	City			Y	0.158193	0.841807
...ate	No	2340	2546	100	360	1	City			Y	0.140894	0.859106
r Graduate	No	3276	0	78	360	1	City			Y	0.329375	0.670625
r Graduate	Yes	2165	3422	152	360	1	City			Y	0.28222	0.71778
r Graduate	No	2226	0	59	360	1	Town			Y	0.272703	0.727297
r Graduate	No	3881	0	147	360	0	Village			N	0.93692	0.06308
...ate	No	13633	0	280	240	1	City			Y	0.131183	0.868817
r Graduate	No	2400	2400	123	360	1	Town			Y	0.236009	0.763991
r Graduate	No	3091	0	90	360	1	City			Y	0.334946	0.665054
...ate	No	2185	1516	162	360	1	Town			Y	0.158905	0.841095
...ate	No	4166	0	40	180	1	City			Y	0.187291	0.812709
...ate	No	12173	0	166	360	0	Town			N	0.789355	0.210645
...ate	No	4666	0	124	360	1	Town			Y	0.14597	0.85403
...ate	No	5667	0	131	360	1	City			Y	0.359743	0.640257
...ate	No	4583	2916	200	360	1	City			Y	0.164788	0.835212
...ate	No	3786	333	126	360	1	Town			Y	0.090288	0.909712
...ate	No	9226	7916	300	360	1	City			Y	0.283118	0.716882
...ate	No	1300	3470	100	180	1	Town			Y	0.141781	0.858219
r Graduate	No	1888	1620	48	360	1	City			Y	0.314745	0.685255
r Graduate	No	2083	0	28	180	1	City			Y	0.252377	0.747623
...ate	No	3909	0	101	360	1	City			Y	0.252618	0.747382
r Graduate	No	3765	0	125	360	1	City			Y	0.341495	0.658505
...ate	No	5400	4380	290	360	1	City			Y	0.251407	0.748593
...ate	No	0	24000	148	360	0	Village			N	0.988813	0.011187
...ate	No	4363	1250	140	360	1	City			Y	0.139004	0.860996
...ate	No	7500	3750	275	360	1	City			Y	0.235474	0.764526
...ate	No	3772	833	57	360	1	Town			Y	0.083689	0.916311
...ate	No	2942	2382	125	180	1	City			Y	0.241894	0.758106
r Graduate	No	2478	0	75	360	1	Town			Y	0.193951	0.806049
...ate	No	6250	820	192	360	1	City			Y	0.145686	0.854314

Prediction of Loan Approval Status
APU,TPM

SME_LOAN_ID_NO=LP001015

NO. OF FAMILY MEMBERS	QUALIFICATION	EMPLOYMENT	MONTHLY INCOME	CO-APPLICANT INCOME	LOAN AMOUNT	LOAN DURATION (MONTHS)	LOAN HISTORY	LOAN LOCATION	LOAN_APPROVAL_STATUS	From: LOAN_APPROVAL_STATUS	Into: LOAN_APPROVAL_STATUS	Predicted Probability: LOAN_APPROVAL_STATUS=N	Predicted Probability: LOAN_APPROVAL_STATUS=Y
0	Graduate	No	5720	0	110	360	1	City			Y	0.1582297	0.8417703

----------End of Report----------

Prediction of Loan Approval Status
APU,TPM

SME_LOAN_ID_NO=LP001022

NO. OF FAMILY MEMBERS	QUALIFICATION	EMPLOYMENT	MONTHLY INCOME	CO-APPLICANT INCOME	LOAN AMOUNT	LOAN DURATION (MONTHS)	LOAN HISTORY	LOAN LOCATION	LOAN_APPROVAL_STATUS	From: LOAN_APPROVAL_STATUS	Into: LOAN_APPROVAL_STATUS	Predicted Probability: LOAN_APPROVAL_STATUS=N	Predicted Probability: LOAN_APPROVAL_STATUS=Y
1	Graduate	No	3076	1500	126	360	1	City			Y	0.2574445	0.7425555

----------End of Report----------

Prediction of Loan Approval Status
APU,TPM

SME_LOAN_ID_NO=LP001031

NO. OF FAMILY MEMBERS	QUALIFICATION	EMPLOYMENT	MONTHLY INCOME	CO-APPLICANT INCOME	LOAN AMOUNT	LOAN DURATION (MONTHS)	LOAN HISTORY	LOAN LOCATION	LOAN_APPROVAL_STATUS	From: LOAN_APPROVAL_STATUS	Into: LOAN_APPROVAL_STATUS	Predicted Probability: LOAN_APPROVAL_STATUS=N	Predicted Probability: LOAN_APPROVAL_STATUS=Y
2	Graduate	No	5000	1800	208	360	1	City			Y	0.1581934	0.8418066

----------End of Report----------

12.3.3 Description

Using SAS ODS, the data scientist would like to generate a report about the output information of the forecasting of the status of loan approval based on the new dataset called LIB78400.TESTING_LAS_PRED_TP078400_DS. If the predicted probability of loan approval were greater than the predicted probability of loan rejection, the predicted outcome would be loan approval and vice versa.

For example, for LP001015, the predicted probability of loan approval, 0.842, was greater than the predicted probability of loan rejection, 0.158, so the predicted loan approval status was ‘Y’ indicating loan approved. For LP001022, the predicted probability of loan approval, 0.743, was greater than the predicted probability of loan rejection, 0.15, so the predicted loan approval status was ‘Y’ indicating loan approved. For LP001031, the predicted probability of loan approval, 0.842, was greater than the predicted probability of loan rejection, 0.158, so the predicted loan approval status was ‘Y’ indicating loan approved.

13.0 Data Visualization

13.1 Introduction

The capability of SAS in performing data visualization is extraordinary and revolutionary. This is because of the ability to transform complicated prediction-based statistics into simple yet elegant and meaningful visuals in the form of different charts such as bar and pie charts. SAS users can construct different graphical charts ranging from the most basic bar charts to complex

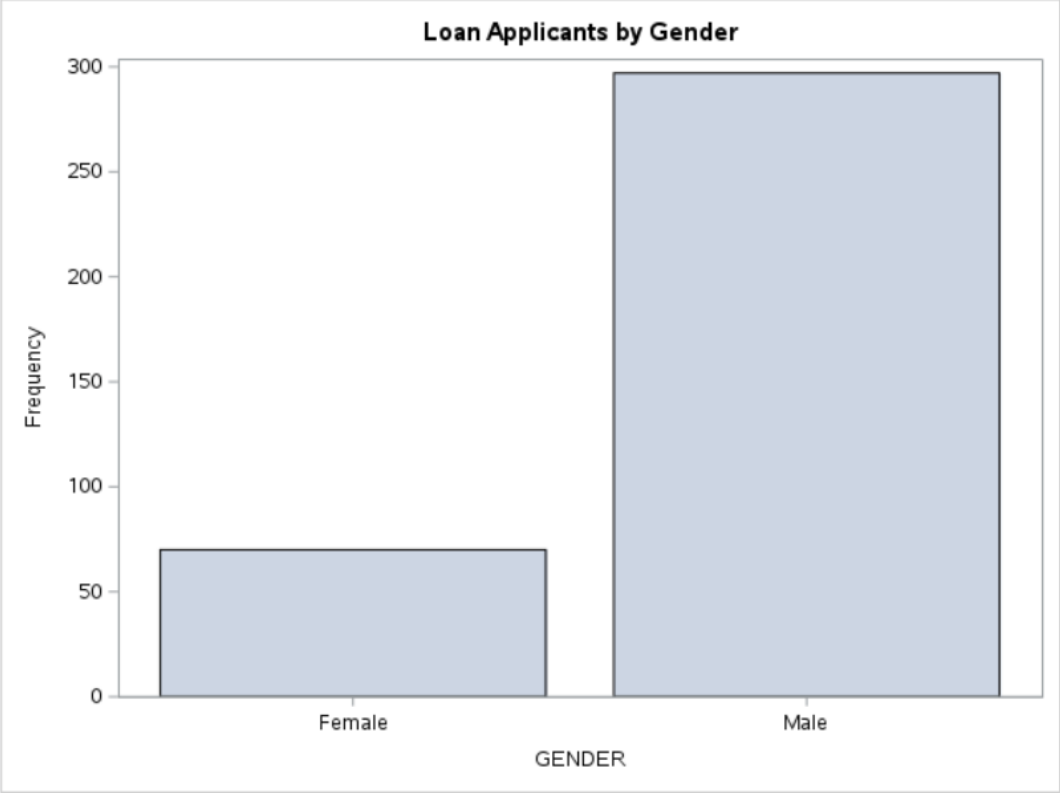
ones to derive more data insights from complex relationships. These insights are always derived in real-time and useful for data scientists and business stakeholders due to SAS's ability to allow users to preprocess data and produce graphic elements with continuous updates. Therefore, data scientists or analysts possess the capability to perform data storytelling that catches the attention of business stakeholders which ultimately facilitates their decision-making process.

13.1.1 Simple Bar Chart

13.1.1.1 SAS Codes

```
1789 /* Data Visualisation Using SAS Codes
1790    Graphical Representation of Information and Data
1791 */
1792
1793 /* SAS Simple Bar Chart */
1794
1795 PROC SGPLOT DATA = LIB78400.TESTING_LAS_PRED_TP078400_DS;
1796 VBAR GENDER;
1797 TITLE 'Loan Applicants by Gender';
1798 RUN;
```

13.1.1.2 Screenshot(s)/Output(s)



13.1.1.3 Description

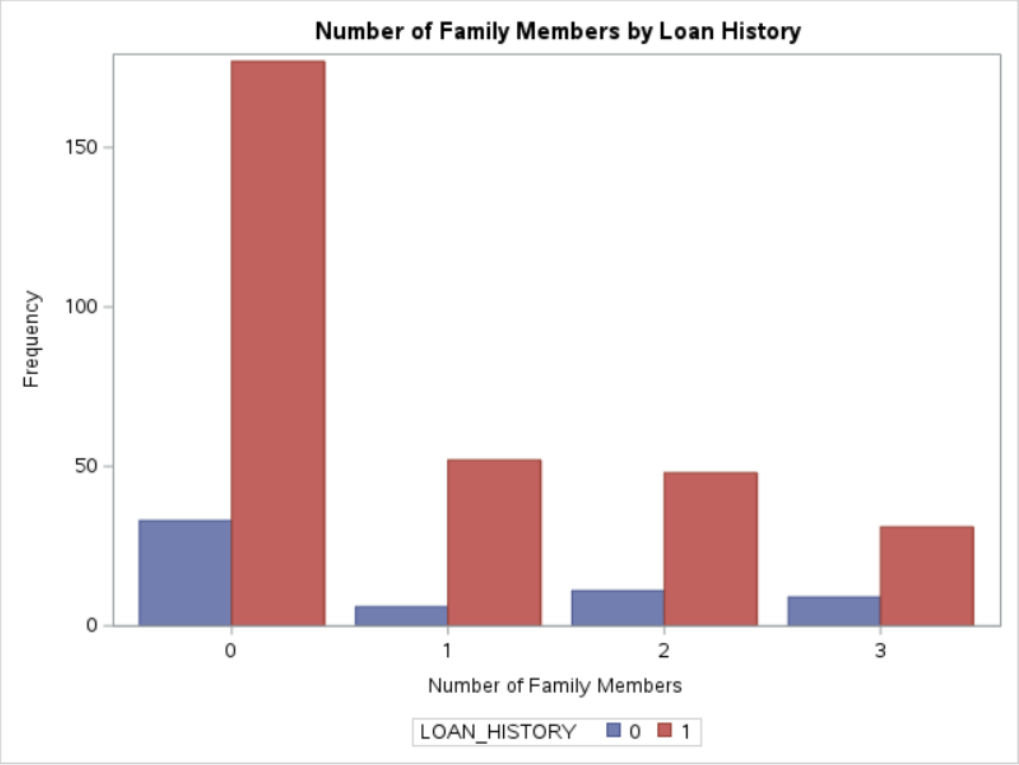
Based on the predicted outcomes of loan approval status, the data scientist visualised the gender distribution of the loan applicants using a simple bar chart. He discovered that the number of male applicants was greater than female applicants.

13.1.2 Stacked Bar Chart

13.1.2.1 SAS Codes

```
1800 /*****************************************
1801 Stacked Bar Chart -
1802 The groups were stacked one above the other
1803 *****************************************/
1804
1805 TITLE 'Number of Family Members by Loan History';
1806 PROC SGPLOT DATA = LIB78400.TESTING_LAS_PRED_TP078400_DS;
1807 VBAR FAMILY_MEMBERS / GROUP = LOAN_HISTORY GROUPDISPLAY = CLUSTER;
1808 Label FAMILY_MEMBERS = 'Number of Family Members';
1809
1810 RUN;
```

13.1.2.2 Screenshot(s)/Output(s)



13.1.2.3 Description

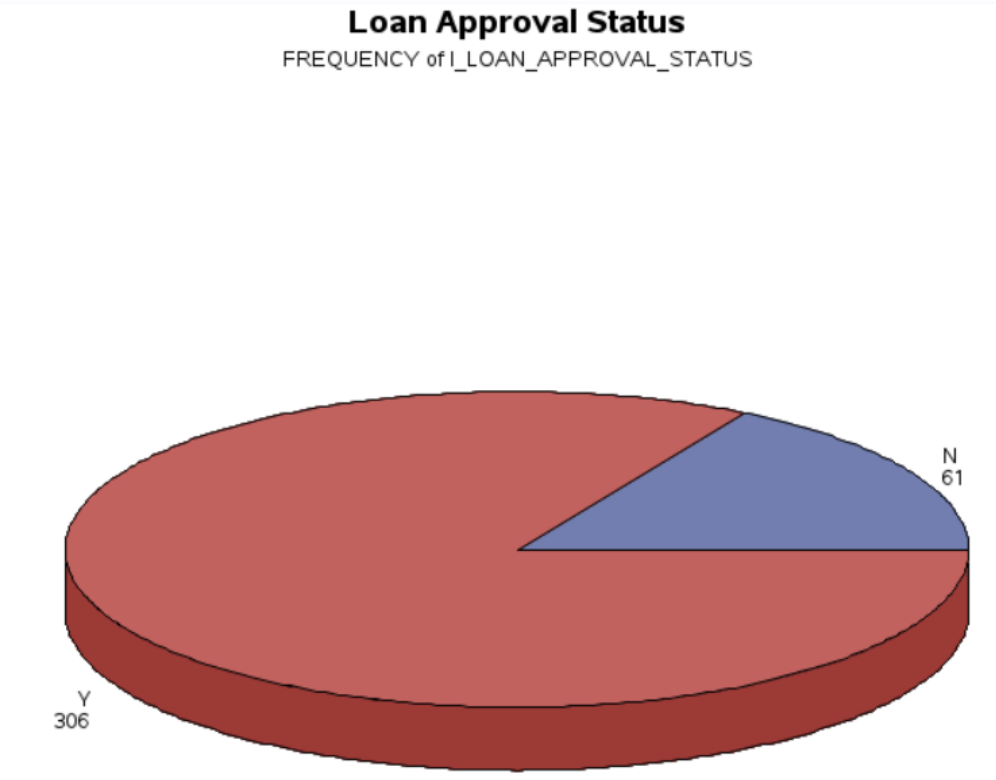
Based on the predicted outcomes of loan approval status, the data scientist visualised the distribution of the number of family members of loan applicants by their loan history using a stacked bar chart. The data scientist discovered that the number of loan applicants with positive loan records was greater than those with negative loan records regardless of the number of family members.

13.1.3 Pie Chart

13.1.3.1 SAS Codes

```
1813 /*************************************************************************************
1814 Pie Chart -
1815 A pie chart is a representation of values as slices of a circle with different colours
1816 *************************************************************************************/
1817
1818 TITLE 'Loan Approval Status';
1819 PROC GCHART DATA = LIB78400.TESTING_LAS_PRED_TP078400_DS;
1820 PIE3D I_LOAN_APPROVAL_STATUS;
1821 RUN;
1822 QUIT;
```

13.1.3.2 Screenshot(s)/Output(s)



13.1.3.3 Description

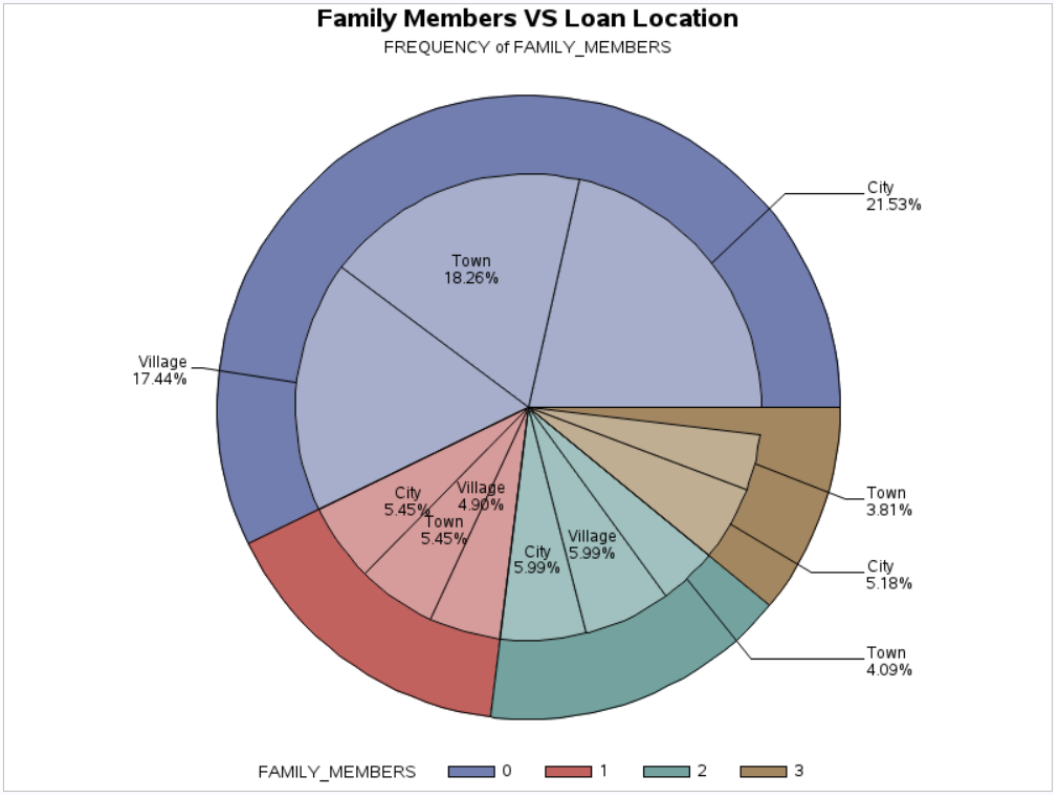
Based on the predicted outcomes of loan approval status, the data scientist visualised the distribution of the loan approval status of loan applicants using a pie chart. The data scientist discovered that the number of loan applicants with approved loans (n = 306) was greater than those with rejected loan applications (n = 61).

13.1.4 Sunburst Chart

13.1.4.1 SAS Codes

```
1824 GOPTIONS RESET = ALL BORDER;
1825 TITLE 'Family Members VS Loan Location';
1826 PROC GCHART DATA = LIB78400.TESTING_LAS_PRED_TP078400_DS;
1827 PIE FAMILY_MEMBERS / DETAIL = LOAN_LOCATION
1828 DETAIL_PERCENT = BEST
1829 DETAIL_VALUE = NONE
1830 DETAIL_SLICE = BEST
1831 DETAIL_THRESHOLD = 2
1832 LEGEND;
1833 RUN;
1834 QUIT;
```

13.1.4.2 Screenshot(s)/Output(s)



13.1.4.3 Description

Based on the predicted outcomes of loan approval status, the data scientist visualised the distribution of the number of family members of loan applicants against loan location using a sunburst chart. The data scientist discovered that 57.23% of loan applicants had no family members, representing the majority of the loan applicant's population. In terms of loan location, most applicants with no family members resided in the city (21.53%), followed by town (18.26%) and village (17.44%).

14.0 Conclusion

It has been an incredible journey for the data scientist in doing and managing this assignment. He was tasked by Lasiandra Finance (LFI) to perform end-to-end loan approval status prediction among loan applicants using the training and testing datasets. He first performed univariate and bivariate analyses of the variables in both datasets to gain a better understanding of the loan applicants' data. He then imputed the missing values of the variables found from the analyses using the mean and mode of the specific continuous and categorical variables respectively so that he could build the best-performing predictive model. He used the logistic regression algorithm as the foundation for developing the model. As a result, he found that loan approval or rejection could be forecasted using significant independent variables including marital status, qualifications as well as loan history and location.

The data scientist had a lot of takeaways from doing this assignment to make his data science journey more interesting and eager to pursue further data science knowledge. He could improve his SAS coding skills throughout this journey so that he could generalize this skill to other end-to-end prediction tasks. Specifically, the use of PROC SQL of SAS in this assignment facilitated his coding process in deriving the necessary insights. This also enhanced his SQL coding skill which was one of the core technical skills for the data science community. Moreover, the documentation process made him realized the importance of professional documentation and how it impresses and benefits clients or business stakeholders in terms of data science professionalism, reading convenience and business decision-making.

The data scientist had to go through toils and snares in performing this assignment. One of the problems faced revolved around exploratory data analysis for the testing set. He performed visualization-oriented manual SAS coding for each of the variables in the training set during the

univariate and bivariate analysis. He could have copy and paste those codes for the testing set but it would significantly increase the length of the coding which might not be convenient and efficient for a data science person, making him a novice in SAS coding. However, Mr Dhason, one of his SAS mentors, taught him how to use SAS macro to solve this problem. This resulted in decreasing the coding length and showcasing his advanced SAS coding skills to others. Therefore, he would like to thank Mr Dhason for his continuous support in using SAS.

Reference

Agarwal, S., & Ben-David, I. (2018). Loan prospecting and the loss of soft information. *Journal of Financial Economics*, *129*(3), 608-628. https://doi.org/10.1016/j.jfineco.2018.05.003

Bansode, N., Verma, A., Sharma, A., & Bhole, V. (2022). Predicting Loan Approval Using ML. *International Research Journal of Modernization in Engineering Technology and Science (IRJMETS)*, *4*(05). https://www.irjmets.com/uploadedfiles/paper//issue_5_may_2022/22186/final/fin_irjmets1651834789.pdf

Bhargav, P., & Sashirekha, K. (2023). A Machine Learning Method for Predicting Loan Approval by Comparing the Random Forest and Decision Tree Algorithms. *Journal of Survey in Fisheries Sciences*, *10*(1S), 1803-1813. https://doi.org/10.17762/sfs.v10i1S.414

Broby, D. (2021). Financial technology and the future of banking. *Financial Innovation*, *7*(1), 47. https://doi.org/10.1186/s40854-021-00264-y

Campbell, D., Loumioti, M., & Wittenberg-Moerman, R. (2019). Making sense of soft information: Interpretation bias and loan quality. *Journal of Accounting and Economics*, *68*(2-3), 101240. https://doi.org/10.1016/j.jacceco.2019.101240

Cetin, H. (2019). The relationship between non-performing loans and selected EU members banks profitabilities. *International Journal of Trade, Economics and Finance*, *10*(2), 52-55. https://www.academia.edu/download/106369982/637-FM015.pdf

Chen, C. (2024). Underwriting and Credit Scoring. In *Practical Credit Risk and Capital Modeling, and Validation: CECL, Basel Capital, CCAR, and Credit Scoring with Examples* (pp. 319-387). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-52542-1_7

Chudappa, S., Thorat, C., Varade, S., & Bhalgaonkar, S. (2023). Loan Approval System using Machine Learning Algorithm. *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering*, *11*(6). https://doi.org/10.17148/ijireeice.2023.11610

Gopinath, M., Maheep, K. S. S., & Sethuraman, R. (2021). Customer loan approval prediction using logistic regression. *Advances in Parallel Computing*. https://ebooks.iospress.nl/pdf/doi/10.3233/APC210103

Heller, D., Leitzinger, L., & Walz, U. (2024). Intellectual Property as Business Loan Collateral: A Taxonomy of Institutional and Economic Determinants. *GRUR International*, *73*(5), 379-392. https://doi.org/10.1093/grurint/ikae043

Johnson, A. M., Villanova, D., & Smith, R. J. (2023). Loan Amount versus Monthly Payments: The Effect of Loan Application Formats on Consumer Borrowing Decisions. *Journal of Consumer Research*, *50*(4), 765-786. https://doi.org/10.1093/jcr/ucad015

Kgoroeadira, R., Burke, A., & van Stel, A. (2019). Small business online loan crowdfunding: who gets funded and what determines the rate of interest?. *Small Business Economics*, *52*, 67-87. https://doi.org/10.1007/s11187-017-9986-z

Madaan, M., Kumar, A., Keshri, C., Jain, R., & Nagrath, P. (2021). Loan default prediction using decision trees and random forest: A comparative study. In *IOP Conference Series: Materials Science and Engineering* (Vol. 1022, No. 1, p. 012042). IOP Publishing. https://doi.org/10.1088/1757-899X/1022/1/012042

Miller, T. (2023, April 17). *7 Successful Debt Collection Techniques to Reduce Bad Debts*. HighRadius Resource Center. Retrieved June 12, 2024, from https://www.highradius.com/resources/Blog/effective-debt-collection-techniques/

Nureni, A. A., & Adekola, O. E. (2022). Loan approval prediction based on machine learning approach. *FUDMA Journal of Science*, *6*(3), 41-50. https://doi.org/10.33003/fjs-2022-0603-830

Orji, U. E., Ugwuishiwu, C. H., Nguemaleu, J. C., & Ugwuanyi, P. N. (2022, April). Machine learning models for predicting bank loan eligibility. In *2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development (NIGERCON)* (pp. 1-5). IEEE. https://doi.org/10.1109/NIGERCON54645.2022.9803172

Sheikh, M. A., Goel, A. K., & Kumar, T. (2020). An approach for prediction of loan approval using machine learning algorithm. *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*. IEEE, 2020. https://doi.org/10.1109/ICESC48915.2020.9155614

Sun, Q., Wang, J., Zhang, H., & Wen, T. (2023). How loan descriptions affect the likelihood that borrowers obtain loans in P2P networks? -An empirical analysis based on the" Renrendai" platform. *Plos One*, *18*(9), e0283508. https://doi.org/10.1371/journal.pone.0283508

Tejaswini, J., Kavya, T. M., Ramya, R. D. N., Triveni, P. S., & Maddumala, V. R. (2020). Accurate loan approval prediction based on machine learning approach. *Journal of Engineering Science*, *11*(4), 523-532. https://jespublication.com/upload/2020-110471.pdf

Volkova, N., Volkova, V., Ptashchenko, L., & Glushko, A. (2019, September). Strategic ways of minimizing banks' bad loan debts. In *6th International Conference on Strategies, Models and Technologies of Economic Systems Management (SMTESM 2019)* (pp. 33-37). Atlantis Press. https://doi.org/10.2991/smtesm-19.2019.8

Yao, J., Chen, J., Wei, J., Chen, Y., & Yang, S. (2019). The relationship between soft information in loan titles and online peer-to-peer lending: evidence from RenRenDai platform. *Electronic Commerce Research*, *19*, 111-129. https://doi.org/10.1007/s10660-018-9293-z

Zhu, L., Qiu, D., Ergu, D., Ying, C., & Liu, K. (2019). A study on predicting loan default based on the random forest algorithm. *Procedia Computer Science*, *162*, 503-513. https://doi.org/10.1016/j.procs.2019.12.017