

**PERSONAL LOAN MODELING USING DATA MINING MODELS:
DECISION TREE, LOGISTIC REGRESSION AND NEURAL NETWORK**

BY

VIVIAN OGECHI OGWUIHE 301200592

A PROJECT

**SUBMITTED IN PARTIAL FUFILLMENT OF THE REQUIREMENTS FOR
BA 706: APPLIED ANALYTIC MODELLING AND
BA 723: BUSINESS ANALYTICS CAPSTONE**

DECEMBER, 2021

TABLE OF CONTENTS

CHAPTER 1.....	INTRODUCTION
1.1.....	Variable or Dataset Dictionary
1.2.....	Data Cleaning
CHAPTER 2.....	DECISION TREE
2.1.....	Decision Tree Modelling for Personal Loan
2.2.....	Misclassification Tree
2.3.....	Average Squared Error
CHAPTER 3.....	REGRESSION
3.1.....	Logistic Regression
3.2	Imputation
3.3	Backward Exclusion Regression Model
3.4	Forward Inclusion Regression Model
3.5	Stepwise Regression
CHAPTER 4.....	NEURAL NETWORKS
	Neural Networks on Imputation Node
	Neural Networks on Transform Node
CHAPTER 5	MODEL COMPARISM
REFERENCES	

CHAPTER 1

INTRODUCTION

Problem Statement

In this project, we are going to use the three major data mining models on the datasets gotten from a US bank - Thera Bank, to solve and answer some business problems and questions respectively, on personal loan.

These problems and questions include:

- a) Predicting whether customers who only come to 'deposit' or 'withdraw' money from the bank will buy a personal loan or not.
- b) In trying to predict the above, which variables are most important?
- c) In terms of employment type, which customers should banks and financial institutions target for large turnover of personal loan sales?
- d) How does Age affect the customer's decision of buying loan?
- e) Do low income earners buy loans?

1.1 Variable or Dataset Dictionary

The file named Bank_Personal_Loan_Modeling.csv contains data on 5000 customers with 14 different variables. The data include customer demographic information (age, income, etc.), the customer's relationship with the bank (mortgage, securities account, etc.), and the customer's response to the last personal loan campaign (Personal Loan). The target variable for this research is the Personal Loan and it is a binary variable which tells us if a customer bought a loan from the bank or not.

The table below contains tabular information on the variables contained in this dataset.

S/N	VARIABLE NAME	DESCRIPTION	LEVELS	ROLE
1	Age	Customer's age in completed Years	Interval	Input
2	CCAvg	Avg.spending on credit cards per month(\$)	Interval	Input
3	CD Account	Does the customer have a certificate of deposit account with the bank?	Binary	Input
4	CreditCard	Does the customer use a credit card issued by this bank?	Binary	Rejected
5	Education	Education Level(1,2,3 for Under graduate, Graduate and Professional/Advanced respectively)	Nominal	Input
6	Experience	Number of Years of professional Experience	Interval	Input
7	Family	Family Size of the Customer	Nominal	Input
8	ID	Customer's ID	Interval	ID
9	Income	Annual Income of the	Interval	Input

		customer(\$)		
10	Mortgage	Value of house mortgage if any(\$)	Interval	Input
11	Online	Does the customer use internet banking facilities?	Binary	Input
12	Personal Loan	Did the customer accept or reject the personal loan offered from last campaign?	Binary	Target
13	Securities Account	Does the customer have a securities account with the bank?	Binary	Input
14	Zip Code	Home Address ZIP code	Interval	Rejected

Table 1: Metadata of the Dataset.

1.2 Data Cleaning

Our data was first imported using the Import Node. It was saved as a SAS file in SAS library. A diagram, data source and library were created under a project named ‘Personal Loan Modelling’ in SAS Enterprise Miner.

There are no missing values in the dataset but there are abnormal values in our datasets. The columns we have are all numerical and personal loan is our target variable. The rest of the variables are categorical.

We do not wish to run our analysis using a variable that has so many categorical values, so the Zip code variable was rejected. Also, we already have average spending on credit card per month, so having Credit Card as a variable will add no new information than the ones we already know. Therefore, to reduce redundancy, CreditCard variable was also rejected.

Variables - EMSave

				(none) ▾	<input type="checkbox"/> not	Equal to	...
Columns: <input type="checkbox"/> Label				<input type="checkbox"/> Mining			
Name	Use	Role	Level				
Age	Default	Input	Interval				
CCAvg	Default	Input	Interval				
CD Account	Default	Input	Binary				
CreditCard	Default	Rejected	Binary				
Education	Default	Input	Nominal				
Experience	Default	Input	Interval				
Family	Default	Input	Nominal				
ID	Default	ID	Interval				
Income	Default	Input	Interval				
Mortgage	Default	Input	Interval				
Online	Default	Input	Binary				
Personal Loan	Default	Target	Binary				
Securities Account	Default	Input	Binary				
ZIP Code	Default	Rejected	Interval				

Fig 1.21 Rejected Variables

Looking at our data on fig 1.22 below, some variables are not having the right datasets. The variable named Experience must be removed as it makes no sense to have '-3' as number of professional experience (Experience).

Sample Statistics

Obs #	Varia...	Label	Type	Percent ...	Minimum	Ma...	Mean
14	ZIP C...	ZIP Co...	VAR	0	9307	96651	93152.5
8	ID		VAR	0	1	5000	2500.5
10	Mortga...		VAR	0	0	635	56.4988
9	Income		VAR	0	8	224	73.7742
1	Age		VAR	0	23	67	45.3384
6	Experi...		VAR	0	-3	43	20.1046
2	CCAvg		VAR	0	0	10	1.9379...
7	Family		VAR	0	1	4	2.3964
5	Educat...		VAR	0	1	3	1.881
3	CD Ac...	CD Ac...	VAR	0	0	1	0.0604
4	Credit...		VAR	0	0	1	0.294
11	Online		VAR	0	0	1	0.5968
12	Person...	Person...	VAR	0	0	1	0.096
13	Securit...	Securit...	VAR	0	0	1	0.1044

Fig 1.22 Sample Statistic Views of Variables

Again, Decision Tree can get influenced by variance and any abnormality in our dataset can change the output we get. It is also almost pointless to have 0 value as the amount of Credit card limit (CCAvg). Therefore, using the Replacement mode, two variables, Average Credit Card spending per month and Number of Years of Experience that had zero and negative

inputs respectively were changed to missing values with the replacement lower limit set to one(1).

Interactive Replacement Interval Filter					
Columns:		<input type="checkbox"/> Label	<input type="checkbox"/> Mining	<input type="checkbox"/> Basic	<input type="checkbox"/> Statistics
Name	Use	Limit Method	Replacement Lower Limit	Replacement Upper Limit	
Age	Default	Default	.	.	
CCAvg	Default	Default	1	.	
Experience	Default	Default	1	.	
Income	Default	Default	.	.	
Mortgage	Default	Default	.	.	
ZIP Code	Default	Default	.	.	

Fig 1.23 Interactive Replacement Editor

When we connected our Statexplore to the replacement node to view our variables, it turned out that SAS rejected all two variables, CCAvg and Experience and to augment this, two new variables, REP_CCAvg and REP_Experience were added as shown in the diagram below.

Statexplore				
Columns: <input type="checkbox"/> Label <input type="checkbox"/> Mining <input type="checkbox"/> Basic <input type="checkbox"/> Statistics				
Name	Use	Report	Role	Level
Age	Default	No	Input	Interval
CCAvg	Default	No	Rejected	Interval
CD Account	Default	No	Input	Binary
CreditCard	Default	No	Rejected	Binary
Education	Default	No	Input	Nominal
Experience	Default	No	Rejected	Interval
Family	Default	No	Input	Nominal
ID	Default	No	ID	Interval
Income	Default	No	Input	Interval
Mortgage	Default	No	Input	Interval
Online	Default	No	Input	Binary
Personal Loan	Default	No	Target	Binary
REP_CCAvg	Default	No	Input	Interval

Fig 1.24 Statexplore window for variables

CHAPTER TWO

DECISION TREE

2.1 Decision Tree Modeling for Personal Loan.

From the Sample icon, we brought in our data partition node into the diagram space and connected it to the Replacement node as shown in fig 2.1 below.

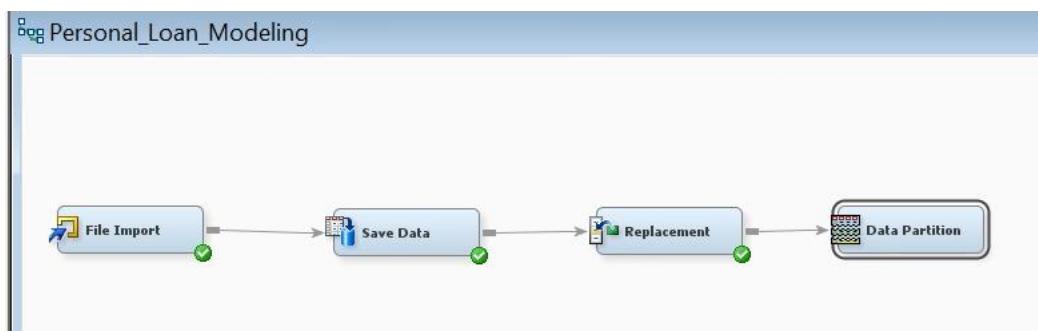


Fig 2.1 Data Partition

On the property panel of data partition node, we allocated equal datasets to our Training and Validation data in fig 2.1. That is, 50 percent of the data was allocated to Validation data, and the remaining 50 percent to out Train data. Our Test data is zero because we do not have enough datasets to give to our Test data as the number of positives on our Target variable(Personal Loan) is far much lower than the number of negatives.

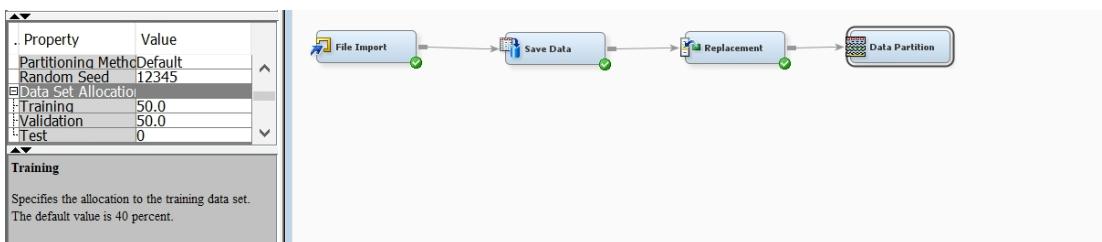


Fig 2.12 Property panel of Data Partition node

We ran our data partition node in fig 2.13 below.

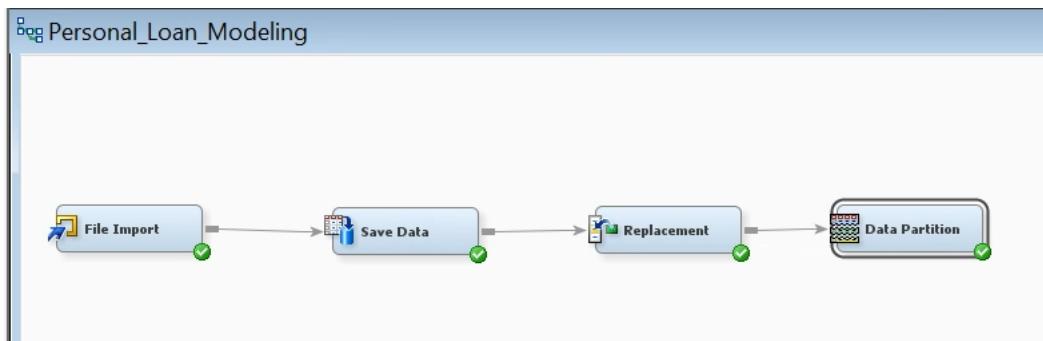


Fig 2.13 Data Partition node (Processed)

Now that we ran our data partition node after we assigned equal number of datasets to our train and validation, we will connect our first Decision Tree to the Data partition node as shown in fig 2.14 below.

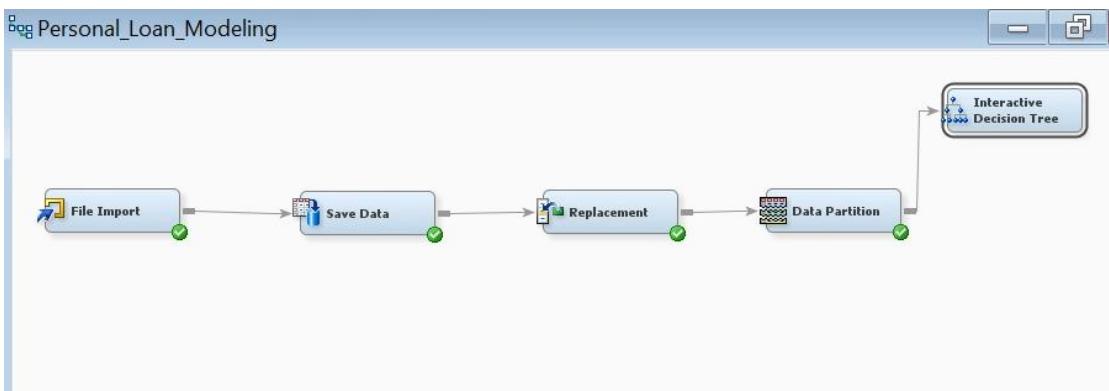


Fig 2.14 Interactive Decision Tree

Unlike other programming methods, SAS is rooted in Statistics and as such, for the first decision tree, we will do most of the adjustments ourselves using the interactive ellipse on the property panel, under the Train section. This way, we can train our data and our Decision Tree will automatically use that analogy to make decision on our validation data.

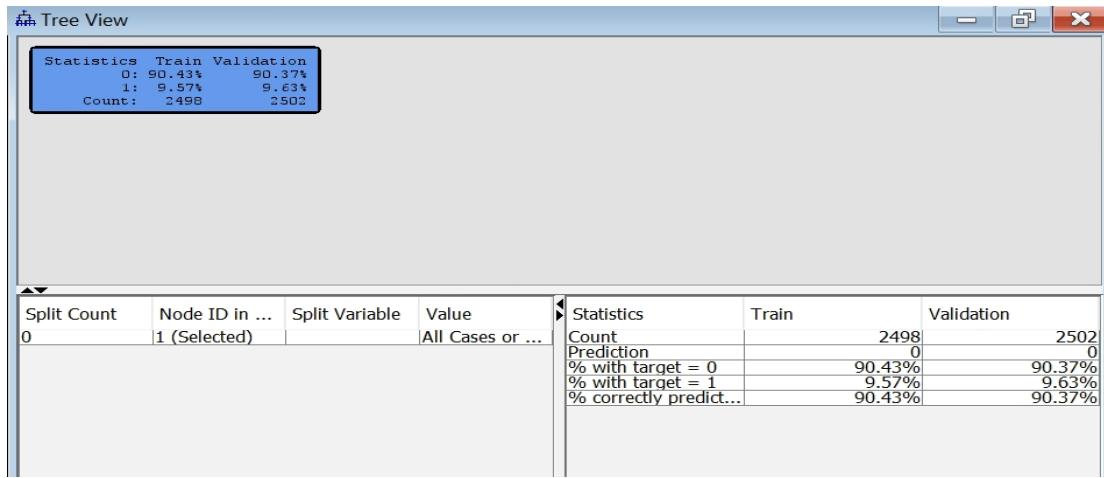


Fig 2.15: Root Node

From our root node or parent node above (highlighted in blue), we can clearly see that the last campaign the bank carried out on personal loan had a lot of negative replies but, compared to previous personal loan campaigns, this result seem to be an improvement as our data source has affirmed, so we will go ahead with the few positives.

According to our root node, 9.57% and 9.63% of customers will buy personal loans from our train and validation datasets respectively.

Let's look at variable importance from our Root Node. Immediately we tried splitting the node, our decision tree gave us the variable according to their importance (highest Log worth).

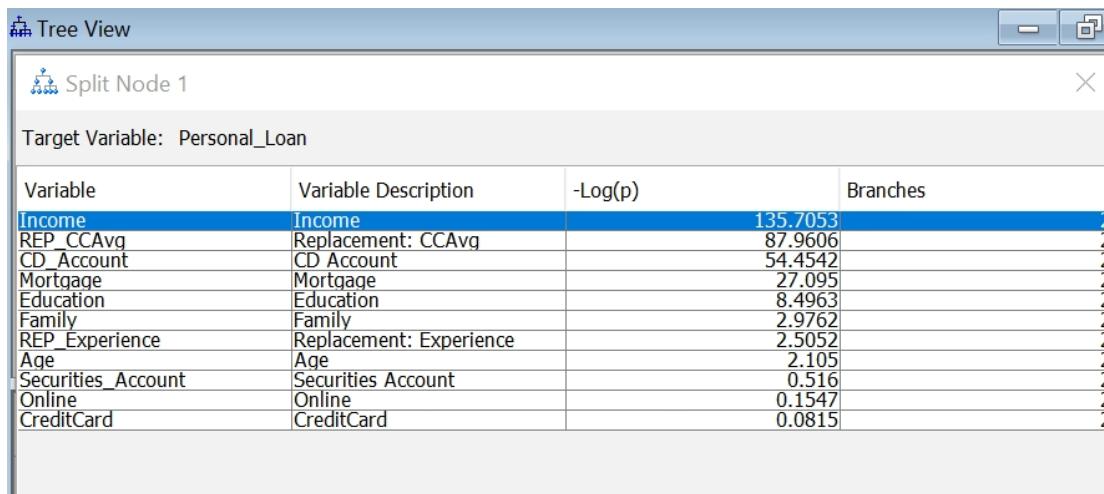


Fig 2.16 Tree View of Root Node for Interactive Tree

In fig 2.16 above, we can see that Income has the highest significance at 135.705 and the least important variable is Credit Card at 0.0815.

We will go ahead and create our Maximal Tree which is the tree with the highest possible number of leaf nodes. From the train node, we created the Maximal Tree which has 13 leaf nodes

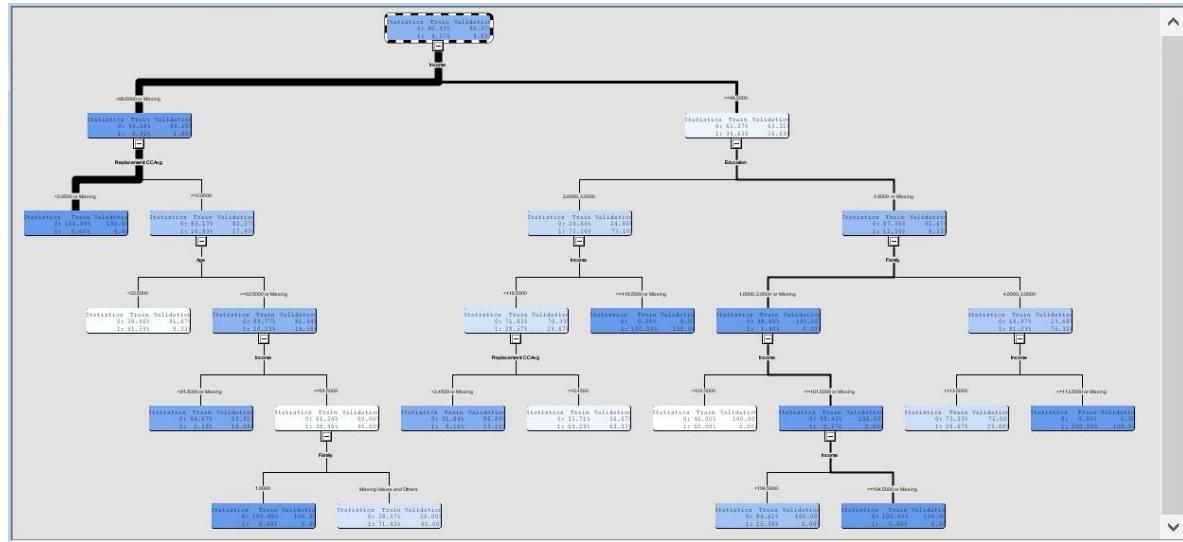


Fig 2.17 Maximal Tree

2.2 Misclassification Tree

We looked at another version of the Decision Tree called Misclassification Tree. We brought in another decision Tree and named it Misclassification Tree then ran it as shown in fig 2.21 below. We want to find out the difference between our predicted outcome and the observed, that is, our misclassification rate.

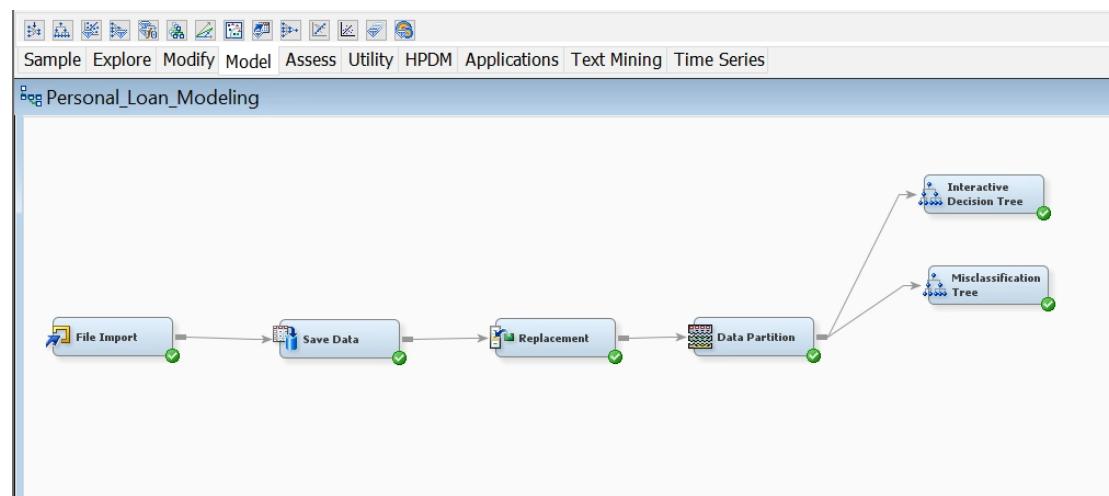


Fig 2.21 Misclassification Tree Node

From the result of our Misclassification Tree below, SAS miner has given us the best number of variables for the best decision according to misclassification. Looking at our leaf statistics, we have 7 leaves compared to the 13 leaves we had in our maximal tree.

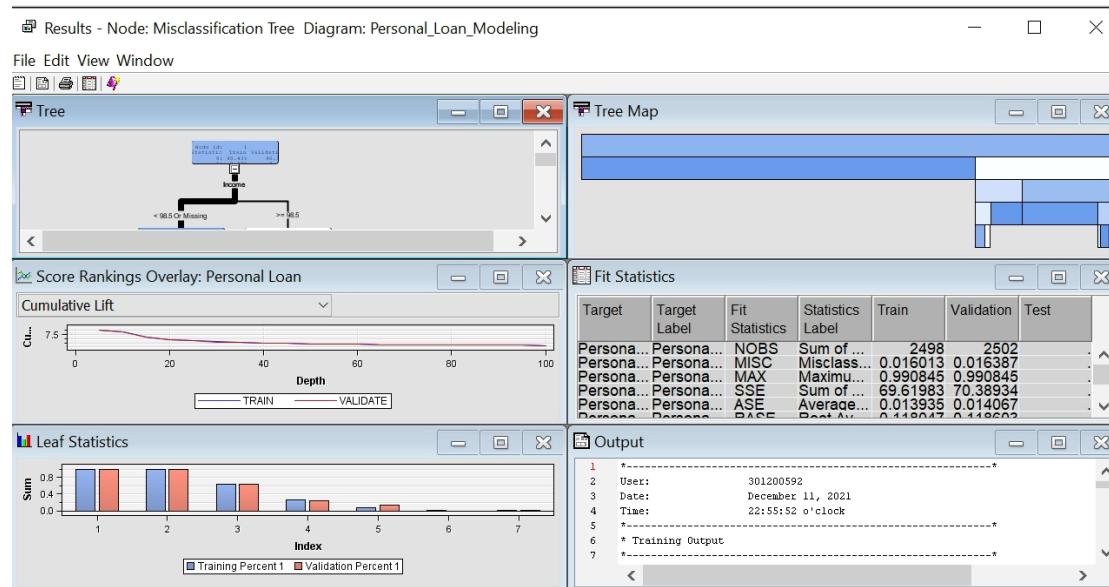


Fig 2.22 Result Window for Misclassification Tree

Just like we had in our maximal tree, our misclassification tree has yet given us Income as the most significant variable in terms of our decision as it has the maximum log worth compared to the other variables.

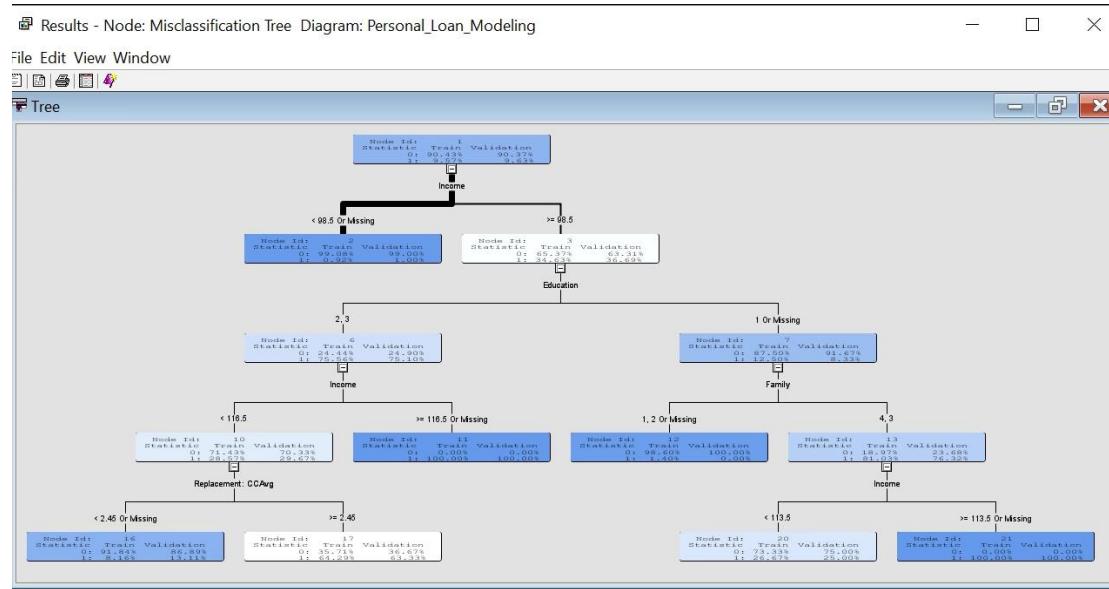


Fig 2.23: Misclassification Tree

Interpreting our Misclassification Tree

From our misclassification Tree, our root node started to split at Income. Bank customers who have exactly income of 98.5 or more have 36.7% chance of buying a personal loan compared to customers who earn less than that income.

Under the category of higher Income earners, customers who are graduates and or advanced in education has 75.1% chance of buying loans compared to customers who are undergraduates or illiterates. The bank should focus more on the first category of customers.

Going forward with customers who have higher education, those who earn 116.5, more or with missing income have 100% chance of buying personal loans compared to the graduates and professionals who earn less than 116.5 of income. The bank should give their maximum effort in marketing these customers with higher education and higher income.

Variable Importance, Misclassification Rate and Average Squared Error (ASE) For Our Misclassification Tree.

According to our Misclassification Tree, we have only four variables that are really very important in our decision. Income, Education, Family and amount of Credit card spent per month, in a decreasing order of importance. They are shown below;

Variable Name	Label	Number of Splitting Rules	Importance	Validation Importance	Ratio of Validation to Training Importance
Income		3	1.0000	1.0000	1.0000
Education		1	0.8217	0.8443	1.0275
Family		1	0.6067	0.4684	0.7720
REP_CCAvq	Replacement: CCAvq	1	0.2555	0.2316	0.9064
Mortgage		0	0.0000	0.0000	.
REP_Experience	Replacement: Experience	0	0.0000	0.0000	.
CreditCard		0	0.0000	0.0000	.
Age		0	0.0000	0.0000	.
CD_Account	CD Account	0	0.0000	0.0000	.
Online		0	0.0000	0.0000	.
Securities_Account	Securities Account	0	0.0000	0.0000	.

Fig 2.24 Variable Importance Window for Misclassification Tree

Also, from the fit statistics window, our misclassification rate depicts that our validation data set actually did a little better than our train data at **0.016387** compared to our train data at **0.016013**, hence, there is no over-fitting. Looking further, our Average error square was **0.013935** and **0.014067** in our train and validation data respectively.

Results - Node: Misclassification Tree Diagram: Personal_Loan_Modeling						
Fit Statistics						
Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
Personal Loan	Personal Loan	NOBS	Sum of Frequencies	2498	2502	.
Personal Loan	Personal Loan	MISC	Misclassification Rate	0.016013	0.016387	.
Personal Loan	Personal Loan	MAX	Maximum Absolute ...	0.990845	0.990845	.
Personal Loan	Personal Loan	SSE	Sum of Squared Err...	69.61983	70.38934	.
Personal Loan	Personal Loan	ASE	Average Squared Er...	0.013935	0.014067	.
Personal Loan	Personal Loan	RASE	Root Average Squar...	0.118047	0.118603	.
Personal Loan	Personal Loan	DIV	Divisor for ASE	4996	5004	.
Personal Loan	Personal Loan	DFT	Total Degrees of Fr...	2498	.	.

Fig 2.25: Fit statistics Window

2.3 Average Squared Error Tree

Again, we ran yet another version of Decision Tree but this time, our assessment measurement was based solely on average squared error (ASE) to produce the result below.

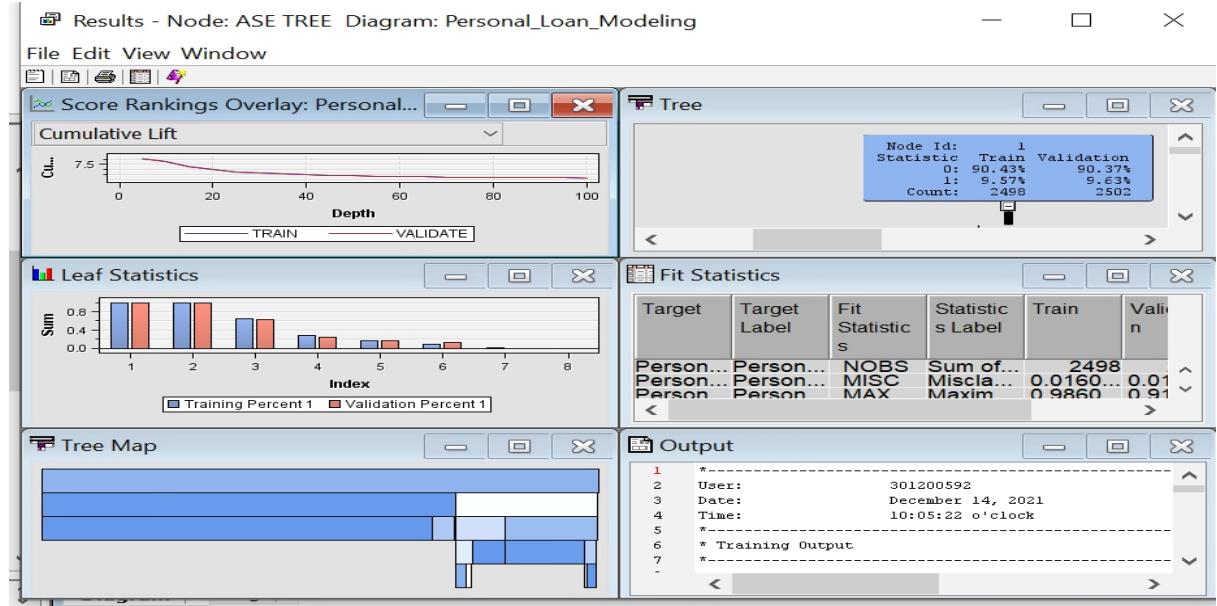


Fig 2.31 ASE Tree result.

Now, let's look at our tree as well as our tree window and fit statistics from the Average Error Tree we have just created.

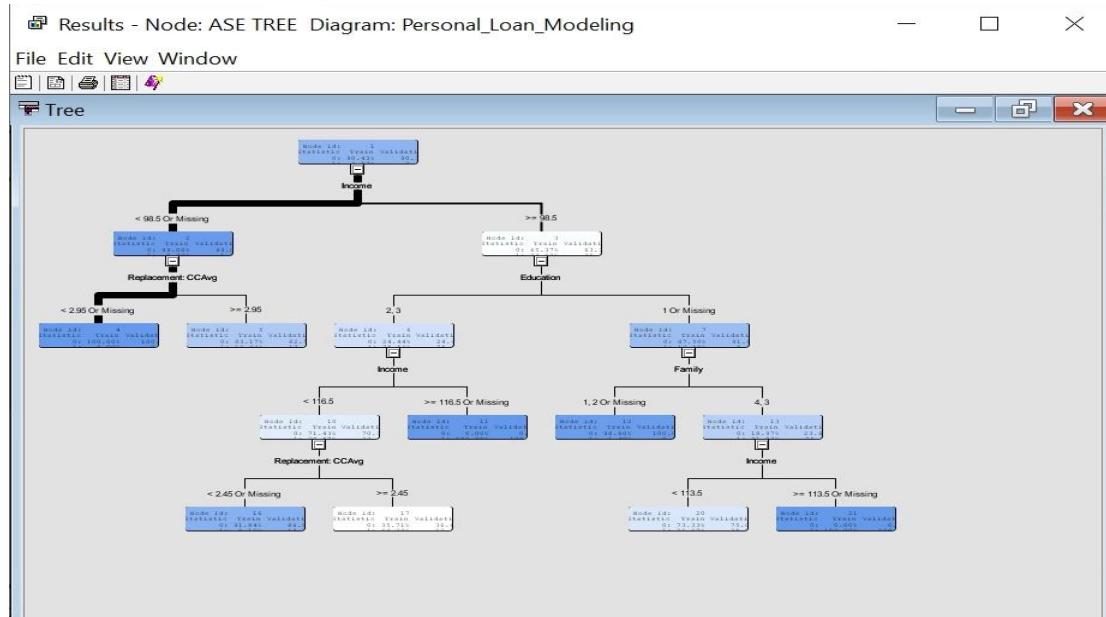


Fig 2.32 Tree window of ASE Decision Tree

The diagram above shows that our ASE Tree produced only 8 leaf nodes with Income as the parent node. For the third time, Income has been proven to be a very strong and important variable in determining whether a customer would buy personal loan or not.

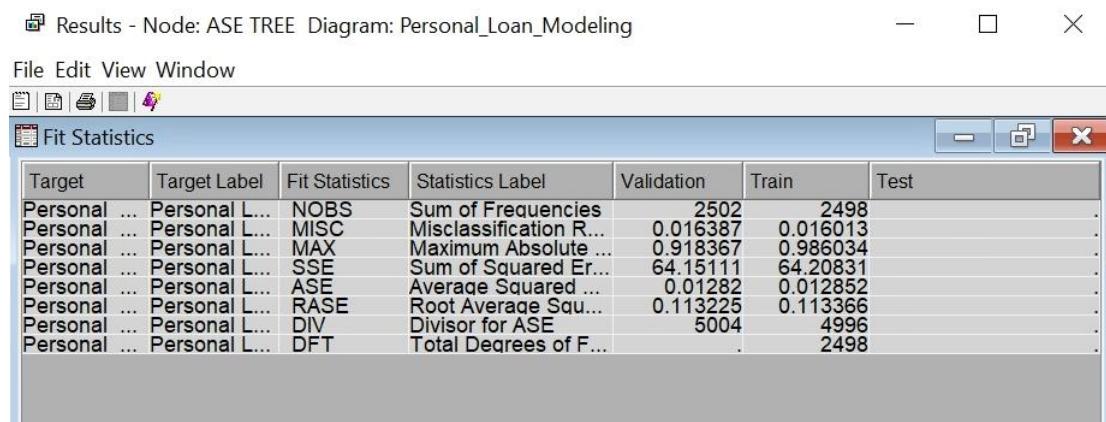


Fig 2.33 Fit Statistic Window of ASE Decision Tree

Clearly, from the fit statistic window above, we did not do very badly for the misclassification rate in our ASE tree for the train and validation data. Likewise, our Average Square Error for both datasets. Average Squared Error for validation data is 0.01282 and that of train data is 0.012852. For the misclassification rate, we have 0.016387 for validation data and 0.16013 for our train data.

All our trees gave us Income as the most important variable but we are yet to decide on the best decision tree. We do this by comparing the average squared error and misclassification rate of the three trees we produced bearing in mind that the best model must not be complex. Therefore, we must go for the less complex model with the lowest error and optimum output.

	Maximal Tree		Misclassification Tree		ASE Tree	
	Val	Train	Val	Train	Val	Train
Average Squared Error	0.17734	0.009649	0.0140	0.0139	0.01282	0.12852
Misclassification Rate	0.020783	0.01201	0.0163	0.0160	0.016387	0.016013
Number of Leaf Nodes Produced	20		7		8	

Table 2: Comparison of the three Decision Trees

Comparing the three decision trees based on misclassification rate and Average Squared Error for our train and validation datasets as seen above, we can decide on the best model for predicting customer's decision on purchasing personal loans. As mentioned earlier, we go for the model with the lowest complexity, minimum error and optimum output. From the table below, our Decision Tree based on average squared Error (ASE Tree) seem to be our best tree with the lowest validation error and eight pure nodes.

CHAPTER THREE

REGRESSION

3.1 Logistic Regression

Unlike Decision Tree, we cannot run our Regression analysis with missing values since we changed some of our zeroes and negative values to missing values when we ran our decision tree. Let us go ahead and explore our data to check for missing values using the StatExplore node connected to our Replacement node as shown in the diagram below.

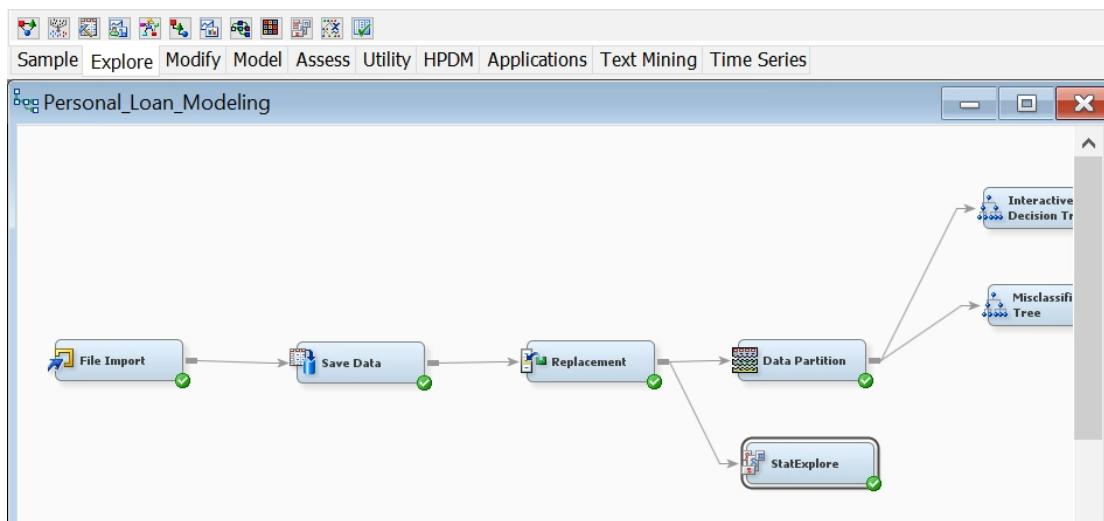


Fig 3.1 Statexplore Node for Viewing our Variables

From the summary statistics of the StatExplore node below, we can see that we have missing values.

Our Rep CCAvg for negative response(0) has 1,636 missing values, REP CCAVG for positive response(1) has 47 missing values, REP Experience for negative response(0) has 111 missing values and REP Experience for positive response(1) has 7 missing values.

Results - Node: StatExplore Diagram: Personal_Loan_Modeling

File Edit View Window

Interval Variables

Data Role	Target	Target Level	Variable	Missing	Median	Non Missing	Minimum	Maximum	Mean
TRAIN	Person... 0		Income	0	59	4520	8	224	66.237
TRAIN	Person... 1		Income	0	142	480	60	203	144.74
TRAIN	Person... 0		Mortgage	0	0	4520	0	635	51.789
TRAIN	Person... 1		Mortgage	0	0	480	0	617	100.84
TRAIN	Person... 0		REP CCAvA	1636	2	2884	1	8.8	2.4619
TRAIN	Person... 1		REP CCAvA	47	4.1	433	1	10	4.2772
TRAIN	Person... 0		REP Experience	111	21	4409	1	43	20.656
TRAIN	Person... 1		REP Experience	7	20	473	1	41	20.137
TRAIN	Person... 0		Age	0	45	4520	23	67	45.367
TRAIN	Person... 1		Age	0	45	480	26	65	45.066

Fig 3.11: Summary Statistics for interval Variables.

3.2 IMPUTATION

Since we have ascertained that we have missing values which will not let our Regression model perform well, we used the imputation node connected to our data partition node below to impute values for our missing data in preparation for regression analysis.

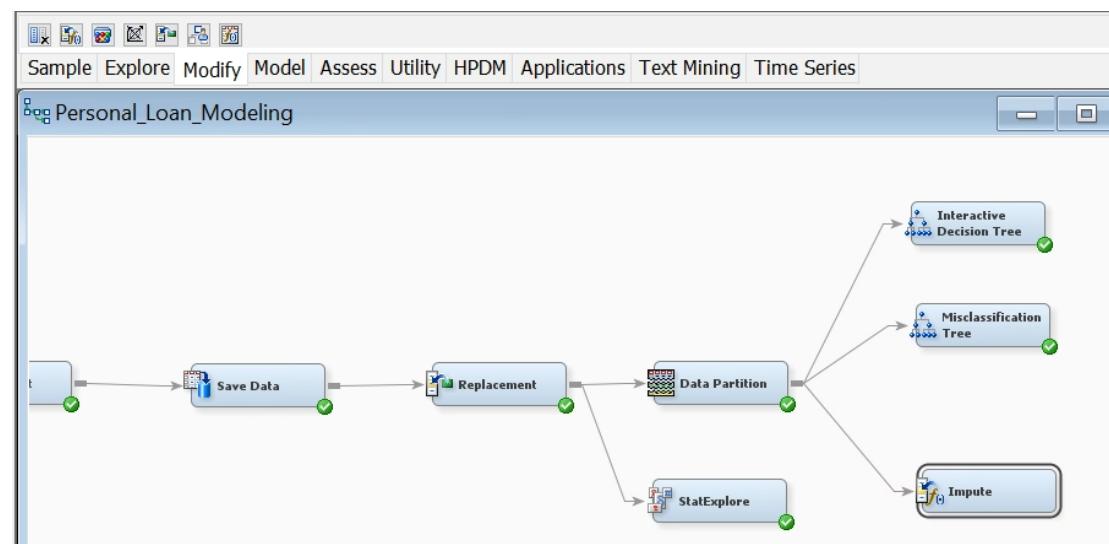


Fig 3.2: Impute Node

On the property panel of our Impute node, we left the default to allow all missing values in class variables to be imputed as mode or as it is called in SAS, count and default impute method for our interval variable as mean. We also flagged our

imputed variables to be unique under the Indicator in the property panel, that way; we will be able to keep a clear track record of all the values that were imputed and be able to differentiate them from our original datasets in the case that there is a need to do so.

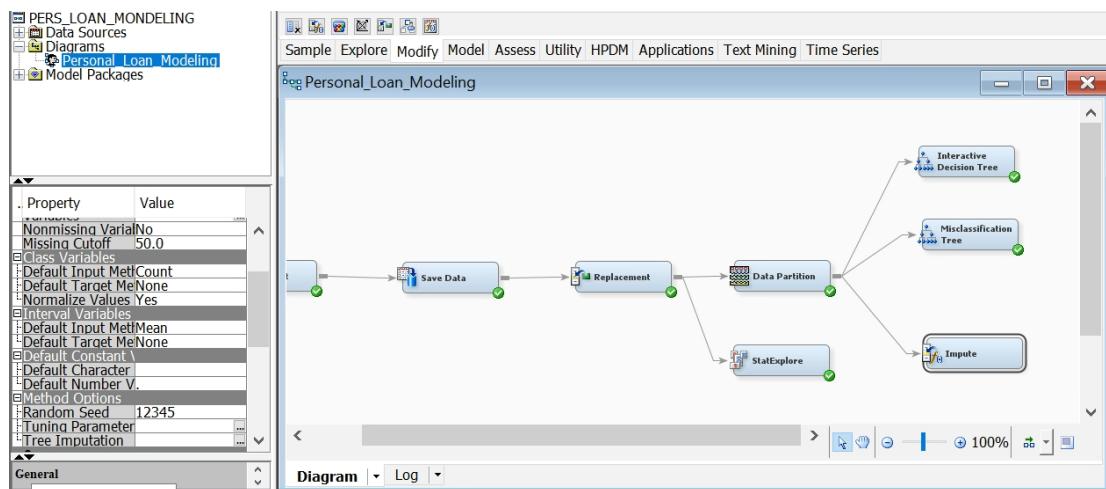


Fig 3.21

Then we ran our Impute Node as shown below.

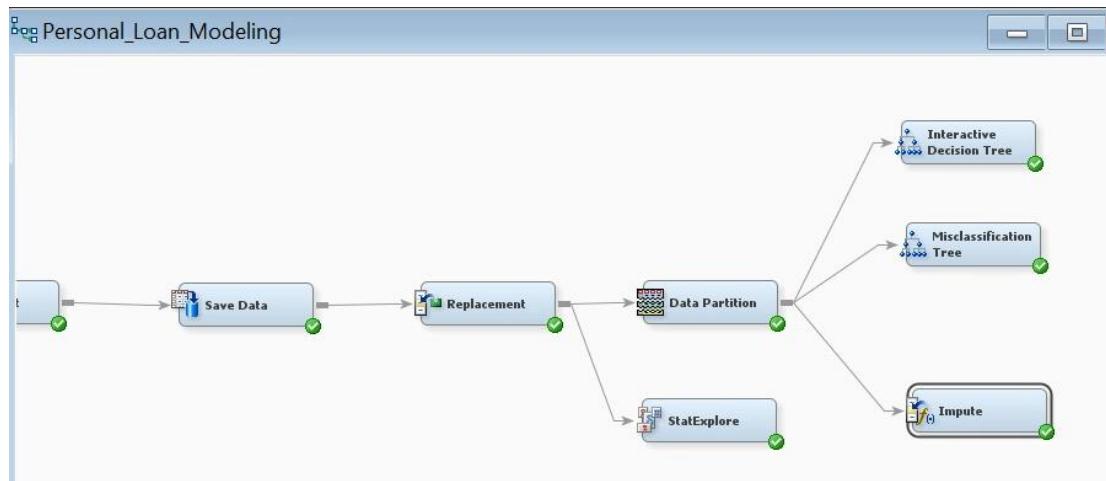


Fig 3.22

As expected from the diagram below, we have a total of two variables that were imputed, namely: IMP_REP_CCAvg and IMP_REP_Experience.

Explore - EMWS1.Impt_VALIDATE

File View Actions Window

EMWS1.Impt_VALIDATE

nt: CCAvg	Imputed: Replacement: Experience	Imputation Indicator for REP_CCAvg	Imputation Indicator for REP_E
1.5	19	0	
1	15	0	
2.714579	13	1	
2.714579	24	1	
2.714579	10	1	
8.9	9	0	
1.5	30	0	
2	27	0	
2.714579	18	1	
3.9	11	0	
2.2	30	0	
1.2	35	0	
2.714579	28	1	
2.714579	6	1	
5	18	0	
1.6	32	0	
2.3	9	0	
1.1	7	0	
2.5	31	0	

Fig 3.23: Imputed variables.

Variable Transformation

Again, in order to make our datasets completely ready for regression, we also have to check for skewness. From the Statexplore node connected to the Impute node, we can see that we have some skewness in some our variables below.

Results - Node: StatExplore Diagram: Personal_Loan_Modeling

File Edit View Window

Interval Variables

Data Role	Target	Target Level	Variable	Skewness	Median	Missing	Non Missing	Minimum	Maximum	Mean	Standard Deviation
TRAIN	Person... 0	Income		0.797544	60	0	2259	8	218	67.0788	46.10
TRAIN	Person... 1	Income		-0.19446	141	0	239	60	203	142.73...	30.41
TRAIN	Person... 0	Mortgage		2.095459	0	0	2259	0	635	51.111...	102.1
TRAIN	Person... 1	Mortgage		1.366181	0	0	239	0	589	108.12...	164.1
TRAIN	Person... 0	IMP REP CCAvg		1.941989	2.714579	0	2259	1	8.8	2.5758...	1.37
TRAIN	Person... 1	IMP REP CCAvg		0.641832	3.7	0	239	1	10.4	4.0262...	1.77
TRAIN	Person... 0	IMP REP Experience		-0.00905	20.61463	0	2259	1	42	20.764...	11.09
TRAIN	Person... 1	IMP REP Experience		0.202142	19	0	239	1	41	19.196	11.7
TRAIN	Person... 0	Age		-0.02926	46	0	2259	23	67	45.555...	11.4
TRAIN	Person... 1	Age		0.225007	43	0	239	26	65	44.0795	11.9

Fig 3.24 Skewness on Statexplore Node

We will go ahead to Log transform the Mortgage and IMP_REP_CCAvg that has outliers in their values using the Transform node connected to the Impute node as shown below.

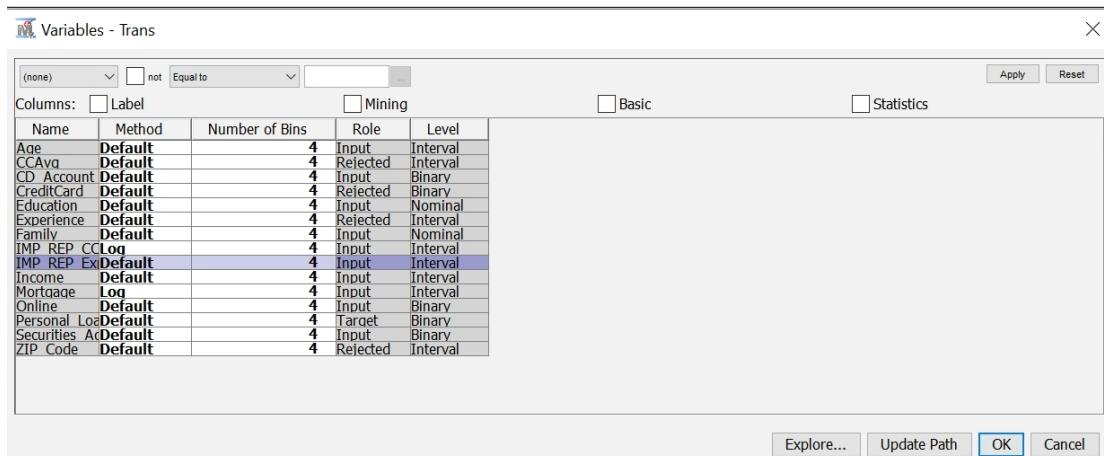


Fig 3.25 Log Transformation of skewed variables.

Interval Variables											
Data Role	Target	Target Level	Variable	Skewness	Median	Missing	Non Missing	Minimum	Maximum	Mean	Standard Deviation
TRAIN Person... 0	Income		0.797544	60	0	2259	8	218	67.0788	46.10	
TRAIN Person... 1	Income		-0.19446	141	0	239	60	203	142.73...	30.48	
TRAIN Person... 0	LOG Mort...		0.891443	0	0	2259	0	6.4551...	1.49824	2.365	
TRAIN Person... 1	LOG Mort...		0.543325	0	0	239	0	6.3801...	2.09753	2.702	
TRAIN Person... 0	LOG IMP ...		0.719393	1.3122...	0	2259	0.6931...	2.2823...	1.2263...	0.317	
TRAIN Person... 1	LOG IMP ...		-0.15367	1.5475...	0	239	0.6931...	2.3978...	1.5526...	0.356	
TRAIN Person... 0	IMP REP ...		-0.00905	20.614...	0	2259	1	42	20.764...	11.05	
TRAIN Person... 1	IMP REP ...		0.202142	19	0	239	1	41	19.196	11.77	
TRAIN Person... 0	Age		-0.02926	46	0	2259	23	67	45.555...	11.41	
TRAIN Person... 1	Age		0.225007	43	0	239	26	65	44.0795	11.9	

Fig 3.26 No skewness

Connecting the Statexplore node to the Transform node, we can see from the above diagram that we no longer have skewness in our data. Therefore, we can proceed with regression analysis with the Regression node connected to the Transform node.

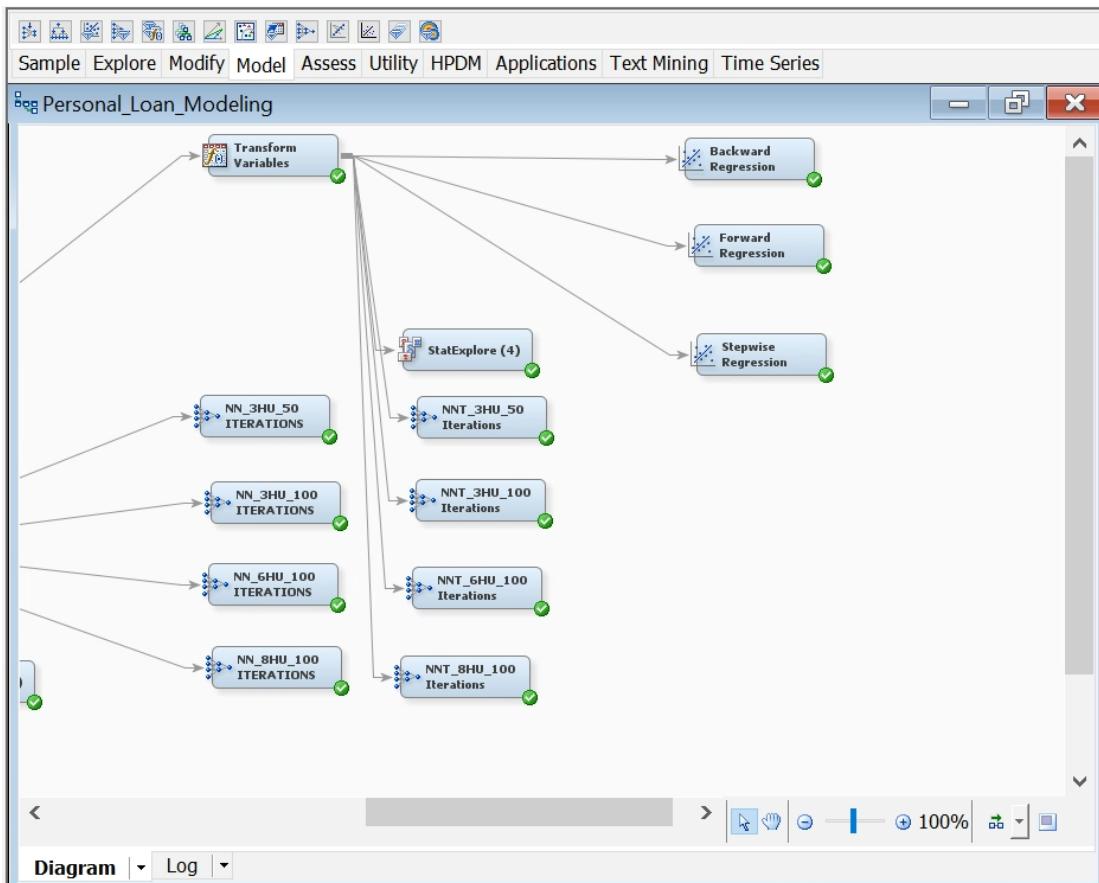


Fig 3.27 All Regression Models connected to the Transform node.

3.3 BACKWARD EXCLUSION REGRESSION MODEL

Before we go any further with the type of Regression we want to run for our analysis, we will run the backward exclusion regression first. This is due to the fact that we do not have many variables, so we do not want to limit the number of variables that will go into our model in the first regression. Therefore, our selection model will be ‘Backward’ from the property panel so that we start our analysis with all our variables, then remove any variable that is not good enough (usually, variable that are not P significant, at P=0.7).

We changed the name of our regression to Backward Exclusion Regression, from the property panel; we also changed our selection criterion to ASE which is Validation Error in Regression. Then we ran it.

Interpretation of Results in Backward Exclusion Regression

A) Fit Statistic Window

Results - Node: Backward Regression Diagram: Personal_Loan_Modeling

File Edit View Window

Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Validation	Train	Test
Personal Loan	Personal Loan	AIC	Akaike's Inf...		664.952	.
Personal Loan	Personal Loan	ASE	Average Sq...	0.028354	0.034014	.
Personal Loan	Personal Loan	AVERR	Average Err...	0.104555	0.127492	.
Personal Loan	Personal Loan	DFE	Degrees of	2484	.
Personal Loan	Personal Loan	DFM	Model Dear...	.	14	.
Personal Loan	Personal Loan	DFT	Total Degre...	.	2498	.
Personal Loan	Personal Loan	DIV	Divisor for ...	5004	4996	.
Personal Loan	Personal Loan	ERR	Error Functi...	523.1948	636.952	.
Personal Loan	Personal Loan	FPE	Final Predic...	.	0.034397	.
Personal Loan	Personal Loan	MAX	Maximum A...	0.999513	0.999732	.
Personal Loan	Personal Loan	MSE	Mean Squa...	0.028354	0.034205	.
Personal Loan	Personal Loan	NOBS	Sum of Fre...	2502	2498	.
Personal Loan	Personal Loan	NW	Number of	14	.
Personal Loan	Personal Loan	RASE	Root Avera...	0.168386	0.184428	.
Personal Loan	Personal Loan	RFPE	Root Final	0.185464	.
Personal Loan	Personal Loan	RMSE	Root Mean ...	0.168386	0.184946	.
Personal Loan	Personal Loan	SBC	Schwarz's	746.4774	.
Personal Loan	Personal Loan	SSE	Sum of Squ...	141.8825	169.9315	.
Personal Loan	Personal Loan	SUMW	Sum of Cas...	5004	4996	.
Personal Loan	Personal Loan	MISC	Misclassific...	0.034772	0.043635	.

Fig 3.31: Fit Statistic Window.

From the Fit statistics window of our regression result above, our Average Squared Error (ASE) for validation data is **0.028354** and our train data is **0.034014**. Also, our Misclassification rate for the validation data is **0.034772** and that of train data is **0.043635**.

B) Output Window

According to our output window, only three variables were removed in our backward exclusion analysis as follows:

First step: Securities_Account was removed, second step: REP_Age got removed

Third step: REP_IMP_Rep_Experience got removed and other remaining variables were considered important or P significant enough (0.05) to remain in our model.

```

505   Education          2 vs 3      0.889
506   Family             1 vs 4      0.163
507   Family             2 vs 4      0.161
508   Family             3 vs 4      1.358
509   Family             0 vs 1      1.259
510   REP_IMP_Rep_CCAvg  1.259
511   REP_Income          1.062
512   REP_Mortgage         1.002
513
514
515   NOTE: No (additional) effects met the 0.05 significance level for removal from the model.
516
517
518   Summary of Backward Elimination
519
520   Step    Effect        Number     Wald      Pr > ChiSq   Validation
521       Removed      DF In      Chi-Square           Error Rate
522
523   1   Securities_Account  1   9      0.0008   0.9780   523.3
524   2   REP_Age            1   8      0.7080   0.4001   524.9
525   3   REP_IMP_Rep_Experience  1   7      0.0702   0.7910   524.0
526
527
528   The selected model, based on the error rate for the validation data, is the model trained in Step 0. It consists of the
529   Intercept CD_Account Education Family Online REP_Age REP_IMP_Rep_CCAvg REP_IMP_Rep_Experience REP_Income REP_
530   Mortgage Securities_Account
531
532
533   Likelihood Ratio Test for Global Null Hypothesis: BETA=0
534
535
536

```

Fig 3.32 Output window of Backward Regression

C) Interpretation of The Odd Ratio Estimates

Effect	Point Estimate	Odds Ratio
CD_Account	0 vs 1	0.073
Education	1 vs 3	0.025
Education	2 vs 3	0.887
Family	1 vs 4	0.158
Family	2 vs 4	0.158
Family	3 vs 4	1.337
Online	0 vs 1	1.704
REP_Age		1.031
REP_IMP_Rep_CCAvg		1.261
REP_IMP_Rep_Experience		0.968
REP_Income		1.062
REP_Mortgage		1.002
Securities_Account	0 vs 1	1.010

Fig 3.33: Odd Ratio Estimates

From the output window above, customers who do not have a certificate of deposit account (CD_Account) have 92.7% chance of not buying a personal loan compared to those that have a certificate of deposit with the bank. A negative correlation with the target variable is seen here.

Looking at the second and third line, customers who are undergraduates and graduates have 97.5% and 11.3% less chances of buying personal loan from the bank compared to customers who have advanced education. This means that education level has a positive correlation with a customer's decision to buy loan. The more education a customer has, the higher chances he/she would purchase a personal loan.

Similarly, customers who have higher number of family members are more likely to purchase personal loan compared to those with fewer family members.

It appears that the more family members the customer has, the more chances the customer will purchase a loan.

REP_Avg and Rep_Income has a positive correlation on loan purchase at 3.1% and 6.2% chances respectively.

The rest of the variables all have positive relationships with personal loan purchase except REP_IMP_REP_Exp which has a negative correlation at 3.2%.

3.4 Forward Inclusion Regression

Our forward Inclusion starts with zero variables or null hypothesis then included P significant variable as it progresses. This type of regression is usually best for datasets with large number of variables as irrelevant variables does not get included when the analysis is run. As usual, our measurement criterion is the average squared error which is Validation Error in regression.

Results - Node: Forward Regression Diagram: Personal_Loan_Modeling

File Edit View Window

Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Validation	Train	Test
Personal Loan	Personal Loan	AIC	Akaike's Infor...		659.7835	
Personal Loan	Personal Loan	ASE	Average Squa...	0.028407	0.034104	
Personal Loan	Personal Loan	AVERR	Average Erro...	0.104717	0.127659	
Personal Loan	Personal Loan	DFE	Degrees of Fr...		2487	
Personal Loan	Personal Loan	DFM	Model Deare...		11	
Personal Loan	Personal Loan	DFT	Total Degree...		2498	
Personal Loan	Personal Loan	DIV	Divisor for ASE	5004	4996	
Personal Loan	Personal Loan	ERR	Error Function	524.0049	637.7835	
Personal Loan	Personal Loan	FPE	Final Predicti...		0.034406	
Personal Loan	Personal Loan	MAX	Maximum Abs...	0.999517	0.999735	
Personal Loan	Personal Loan	MSE	Mean Square...	0.028407	0.034255	
Personal Loan	Personal Loan	NOBS	Sum of Frequ...	2502	2498	
Personal Loan	Personal Loan	NW	Number of Es...		11	
Personal Loan	Personal Loan	RASE	Root Average...	0.168542	0.184674	
Personal Loan	Personal Loan	RFPE	Root Final Pr...		0.185489	
Personal Loan	Personal Loan	RMSE	Root Mean S...	0.168542	0.185082	
Personal Loan	Personal Loan	SBC	Schwarz's Ba...		723.8392	
Personal Loan	Personal Loan	SSE	Sum of Squar...	142.1463	170.3861	
Personal Loan	Personal Loan	SUMW	Sum of Case ...	5004	4996	
Personal Loan	Personal Loan	MISC	Misclassificati...	0.035172	0.043635	

Fig 3.41 Fit Statistic window of Forward Inclusion Regression

From the above window, our average squared error is 0.028407 and 0.034104 for our validation and train data respectively. Also, our misclassification rate was found to be 0.035172 and 0.043635 for our validation and train data respectively.

Results - Node: Forward Regression Diagram: Personal_Loan_Modeling

File Edit View Window

Output

772	REP_IMP_REP_CCAvg	1.259						
773	REP_Income	1.062						
774	REP_Mortgage	1.002						
775								
776								
777	NOTE: No (additional) effects met the 0.05 significance level for entry into the model.							
778								
779	Summary of Forward Selection							
780								
781	Step	Effect Entered	Number DF	In	Score Chi-Square	Pr > ChiSq	Validation Error Rate	
782								
783	1	REP_Income	1	1	582.3424	<.0001	974.2	
784	2	Education	2	2	219.6736	<.0001	648.1	
785	3	Family	3	3	70.4245	<.0001	595.1	
786	4	CD_Account	1	4	52.8924	<.0001	537.5	
787	5	REP_IMP_REP_CCAvg	1	5	8.0427	0.0046	525.8	
788	6	REP_Mortgage	1	6	6.3466	0.0118	530.7	
789	7	Online	1	7	5.6976	0.0170	524.0	
790								
791								
792								
793								
794	The selected model, based on the error rate for the validation data, is the model trained in Step 7. It consists of t							
795								
796	Intercept	CD_Account	Education	Family	Online	REP_IMP_REP_CCAvg	REP_Income	REP_Mortgage
797								
798								
799	Likelihood Ratio Test for Global Null Hypothesis: BETA=0							
800								
801	-2 Log Likelihood	Likelihood						
802								

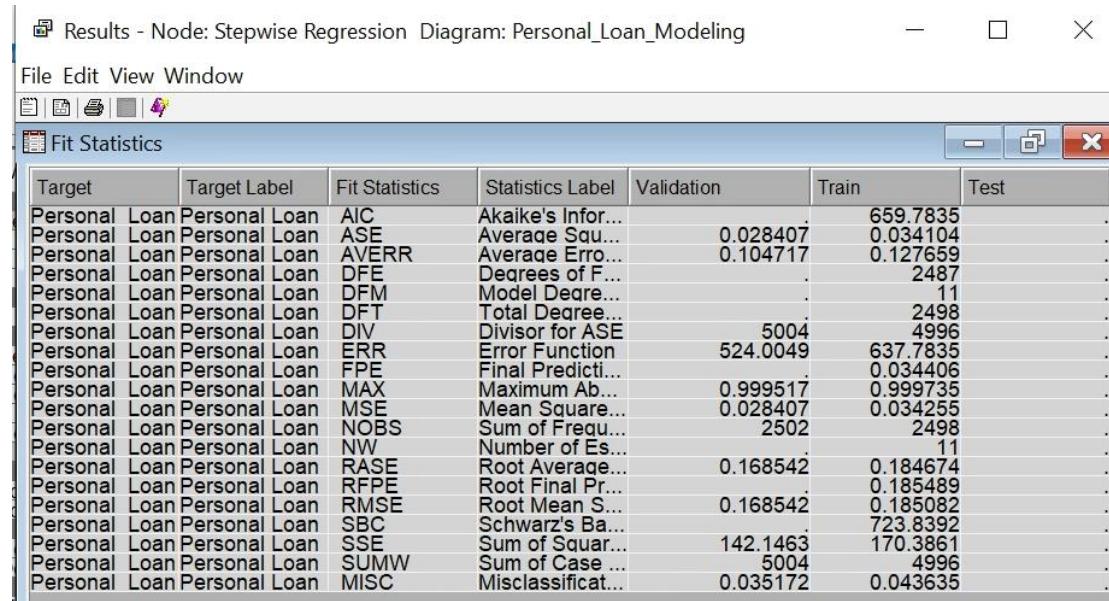
Fig 3.42 Output Window of Forward Inclusion Regression

From the Output window above, we can see that seven (7) variables were included as p significant ($p=0.05$) in our forward inclusion model. With the exception of Intercept that is always present, Income was the first variable to get included and Online (availability of internet enabled device) was the last variable to get included in our model.

3.5 Stepwise Regression

The stepwise regression is a combination of backward and forward regression. It is a balance of the two in the sense that it runs both forward inclusion and backward exclusion analysis on our dataset at the same time. In this project, we also used the stepwise regression model to check variable importance in the prediction of customer's decision to purchase personal loan.

Like we have always done, we connected the stepwise regression model to the impute node and changed our selection criterion to average squared error and our model of regression to stepwise using the property panel of the regression node. Let's look at the fit statistics and output windows of our stepwise regression model below.



The screenshot shows the 'Fit Statistics' window from SPSS. The window title is 'Fit Statistics' and the main title bar says 'Results - Node: Stepwise Regression Diagram: Personal_Loan_Modeling'. The window contains a table with columns: Target, Target Label, Fit Statistics, Statistics Label, Validation, Train, and Test. The table lists various statistical measures for a 'Personal Loan Personal Loan' target, including AIC, ASE, AVERR, DFE, DFM, DFT, DIV, ERR, FPE, MAX, MSE, NOBS, NW, RASE, RFPE, RMSE, SBC, SSE, SUMW, and MISC. The 'Validation' column shows values like 0.028407, 0.104717, etc., while 'Train' and 'Test' columns show values like 659.7835, 0.034104, etc.

Target	Target Label	Fit Statistics	Statistics Label	Validation	Train	Test
Personal	Loan Personal	AIC	Akaike's Infor...		659.7835	
Personal	Loan Personal	ASE	Average Squ...	0.028407	0.034104	
Personal	Loan Personal	AVERR	Average Erro...	0.104717	0.127659	
Personal	Loan Personal	DFE	Degrees of F...		2487	
Personal	Loan Personal	DFM	Model Deare...		11	
Personal	Loan Personal	DFT	Total Deare...		2498	
Personal	Loan Personal	DIV	Divisor for ASE	5004	4996	
Personal	Loan Personal	ERR	Error Function	524.0049	637.7835	
Personal	Loan Personal	FPE	Final Predicti...		0.034406	
Personal	Loan Personal	MAX	Maximum Ab...	0.999517	0.999735	
Personal	Loan Personal	MSE	Mean Square...	0.028407	0.034255	
Personal	Loan Personal	NOBS	Sum of Frequ...	2502	2498	
Personal	Loan Personal	NW	Number of Es...		11	
Personal	Loan Personal	RASE	Root Average...	0.168542	0.184674	
Personal	Loan Personal	RFPE	Root Final Pr...		0.185489	
Personal	Loan Personal	RMSE	Root Mean S...	0.168542	0.185082	
Personal	Loan Personal	SBC	Schwarz's Ba...		723.8392	
Personal	Loan Personal	SSE	Sum of Squar...	142.1463	170.3861	
Personal	Loan Personal	SUMW	Sum of Case ...	5004	4996	
Personal	Loan Personal	MISC	Misclassificat...	0.035172	0.043635	

Fig 3.51 Fit Statistic window of Stepwise Regression Model

The screenshot shows the SPSS Output window titled "Results - Node: Stepwise Regression Diagram: Personal_Loan_Modeling". The window displays the following text:

```

775
776
777 NOTE: No (additional) effects met the 0.05 significance level for entry into the model.
778
779
780 Summary of Stepwise Selection
781
782      Effect          Number       Score        Wald      Validation
783      Step    Entered     DF   In   Chi-Square   Chi-Square   Pr > ChiSq   Error Rate
784
785      1 REP_Income      1   1   582.3424           <.0001    974.2
786      2 Education       2   2   219.6736           <.0001    648.1
787      3 Family          3   3   70.4245           <.0001    595.1
788      4 CD_Account      1   4   52.8924           <.0001    537.5
789      5 REP_IMP_REP_CCAvg 1   5   8.0427            0.0046    525.8
790      6 REP_Mortgage     1   6   6.3466            0.0118    530.7
791      7 Online          1   7   5.6976            0.0170    524.0
792
793
794 The selected model, based on the error rate for the validation data, is the model trained in Step 7. It consists of t
795 Intercept CD_Account Education Family Online REP_IMP_REP_CCAvg REP_Income REP_Mortgage
796
797
798 Likelihood Ratio Test for Global Null Hypothesis: BETA=0
799
800 -2 Log Likelihood      Likelihood
801 Intercept Intercept &      Ratio
802 Only Covariates      Chi-Square     DF      Pr > ChiSq
803
804

```

Fig 3.52 Output window of Stepwise Regression.

We noticed that both our Stepwise and Forward Regression models gave us the same output. Now we compare these three models in the table below.

	Backward Inclusion Regression		Forward Inclusion Regression		Stepwise Regression	
	Val	Train	Val	Train	Val	Train
Average Squared Error	0.028354	0.034014	0.028407	0.034104	0.028407	0.034104
Misclassification Rate	0.034772	0.043635	0.036771	0.043635	0.035172	0.043635
P Significant variables	7		6		7	

Table 3 Comparison of our three Regression Models

From the table above, not only does our backward exclusion regression have the lowest average squared error, it has the lowest misclassification rate. Therefore, our best model in the logistic regression so far is the **Backward Exclusion model** of our Regression analysis.

CHAPTER FOUR

NEURAL NETWORK

A neural network is a series of algorithms that endeavours to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature. (Investopia, 2021).

We are going to run our datasets on neural network using different hidden units and iterations.

Like Regression, Neural network does not do well with missing values, therefore, we would run our neural network on the imputed node.

Even much better is neural network that is modeled on other predictive model. Since our Decision Tree model has missing values, we would go with Regression and from the earlier pages; Backward Exclusion Regression seems to be our best shot. Components of a simple neural network comprises of an input layer, hidden layer and an output layer. We feed our data to the input layer, our datasets get worked on with different mathematical equations inside the hidden layer and the results are produced at the output layer. However, one beautiful thing about neural network is that our output data gets fed into the system again and again until a ‘near’ perfect result is produced. With all the deep learning neural network offers, we are inclined to believe that it will give us the best prediction of the variables that most influences customer’s decision to purchase a personal loan. We would find out in the coming pages.

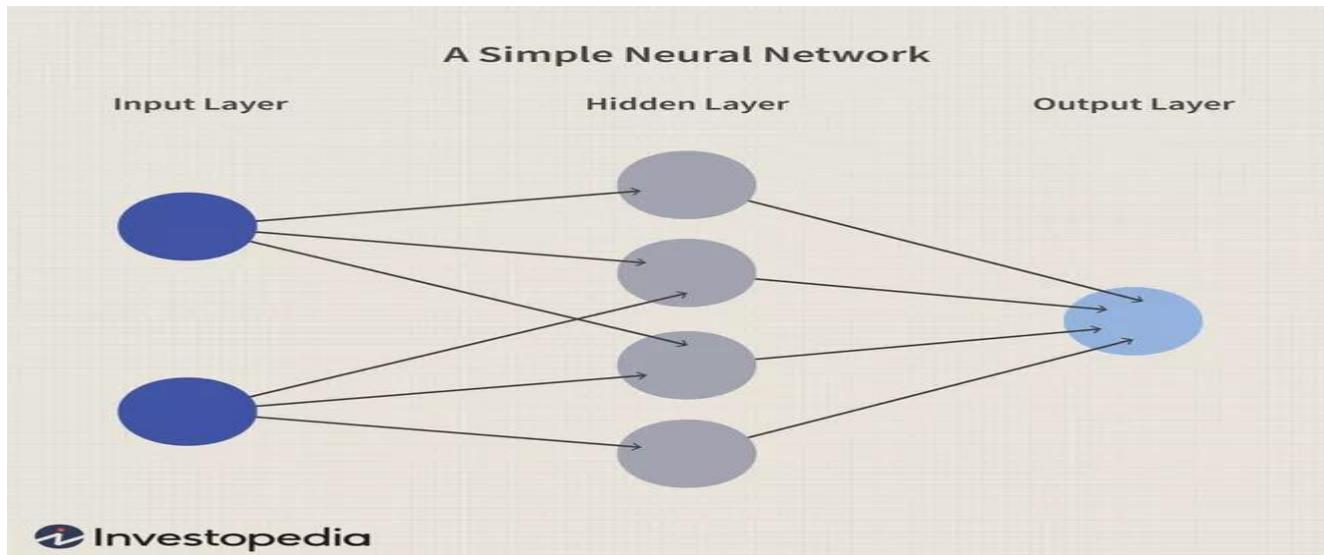


Image by Sabrina Jiang, Investopedia 2021

NEURAL NETWORKS ON IMPUTATION NODE (NN)

1) NN of 3 Hidden Units and 50 Iterations

Firstly, we ran an ordinary neural network connected to the impute node with Average Error as our model selection criterion, 50 iterations, 3 hidden units at default and set our preliminary Test option from the optimization in the Property panel to a No.

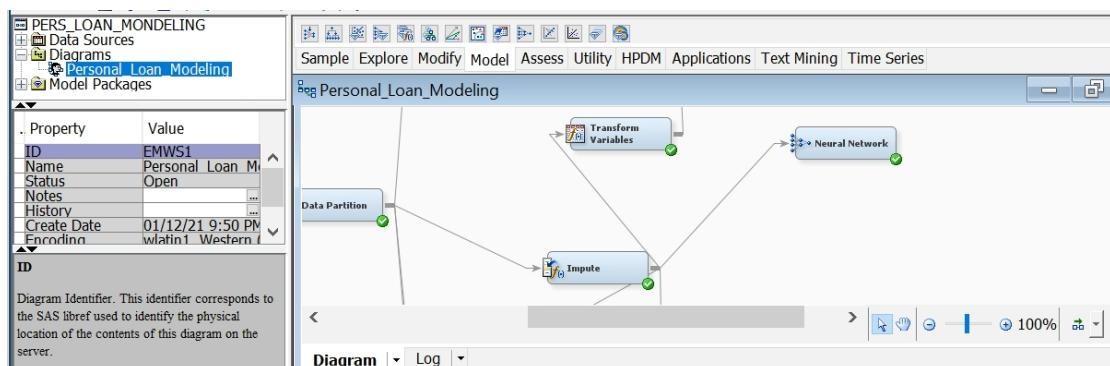


FIG 4.1 Neural Network with 3 hidden Units and 50 iterations.

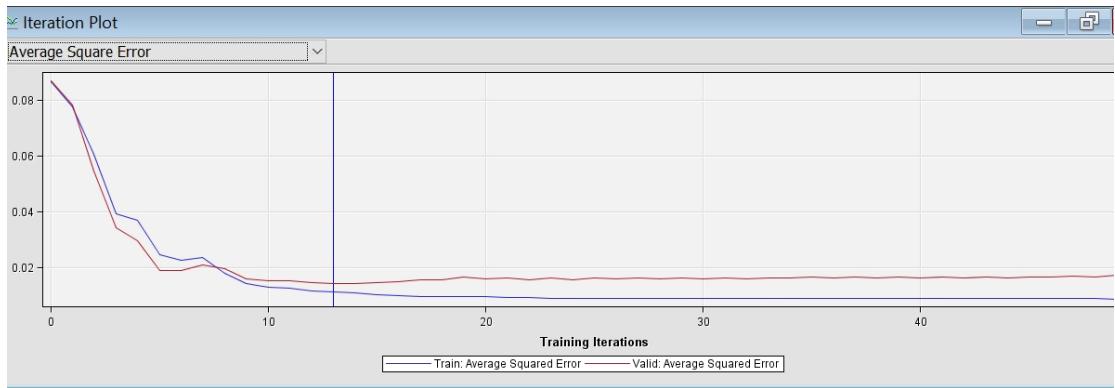


Fig 4.2 Iteration Plot of ASE for NN_3HU with 50 iterations

As seen above, our data converged very early in the iteration and at about the 13th iterations, our data began to degrade for the three hidden unit model (NN_3HU).

2) NN of 3 Hidden Units and 100 Iterations

Again, we built another neural network model with the same settings at for the first one only that this time, we will go with 100 iterations.

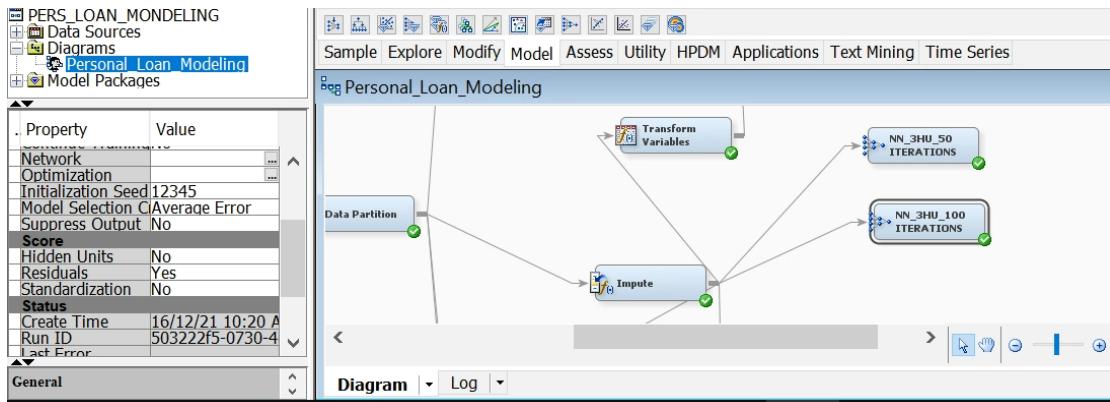


Fig 4.3 NN of 3 Hidden Units and 100 Iterations

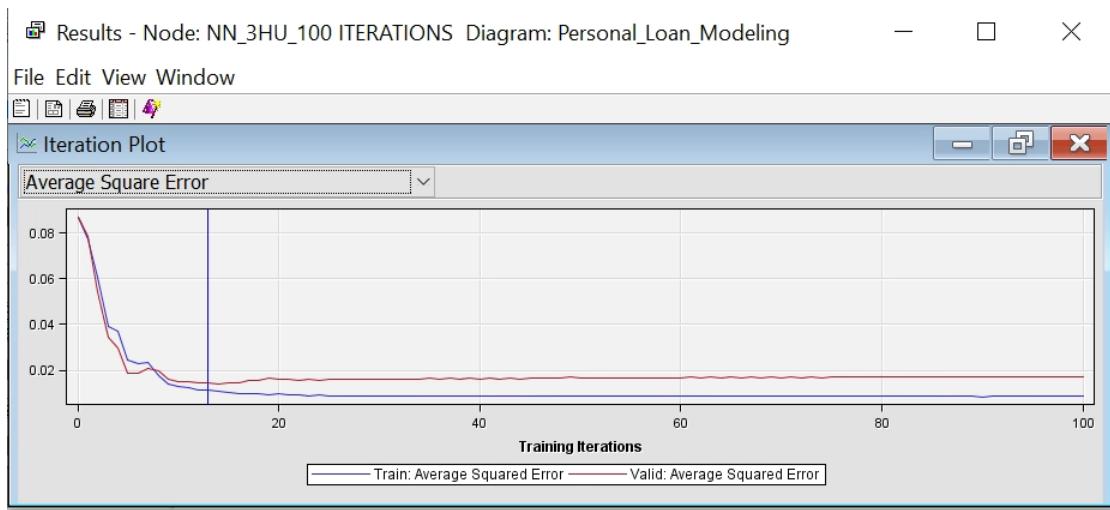


Fig 4.4 Iteration Plot of ASE FOR NN_3HU with 100 Iterations

From the above, it may seem as though we do not need that much of iterations to get our result because, as early as about the 8th iteration, our model began to degrade after converging from the beginning.

It can also be deduced the more the hidden units, the more convergence we get from the first iterations and the model degrades shortly after some little iteration.

3) Neural Network of 6 Hidden Unit and 100 Iterations

Now we try another model with 100 iterations and 6 hidden units.

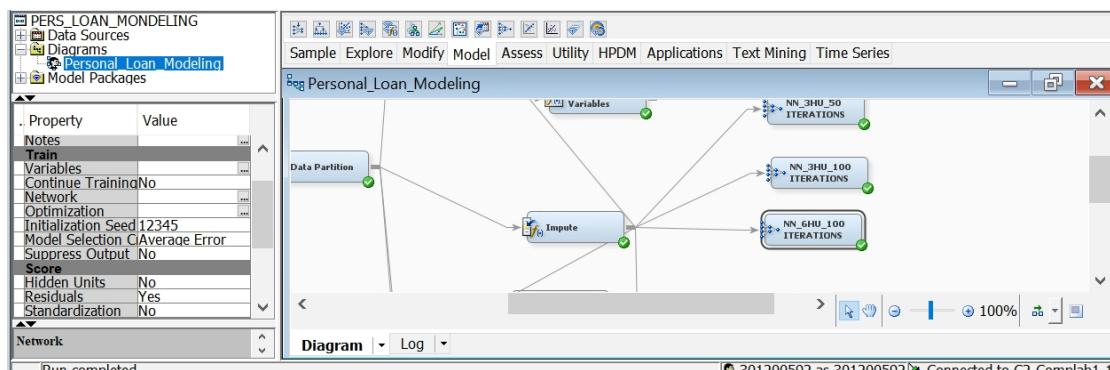


Fig 4.5 NN_6HU_100 ITERATIONS

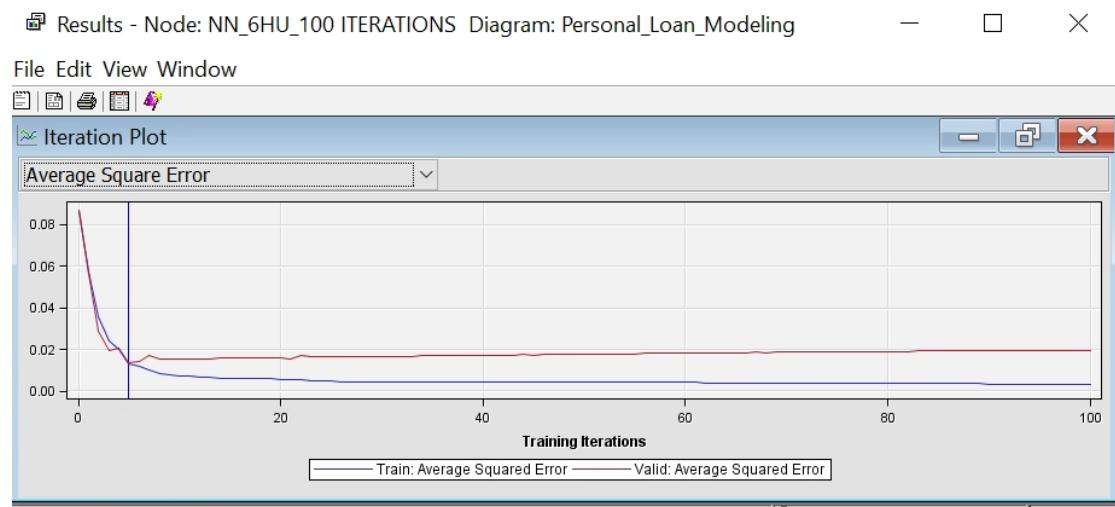


Fig 4.6 Iterative Plot of ASE of NN_6HU_1000 Iterations.

4) Neural Network of 8 Hidden Unit and 100 Iterations

Going further, we would look at neural network of 8 hidden units with 100 iterations below:

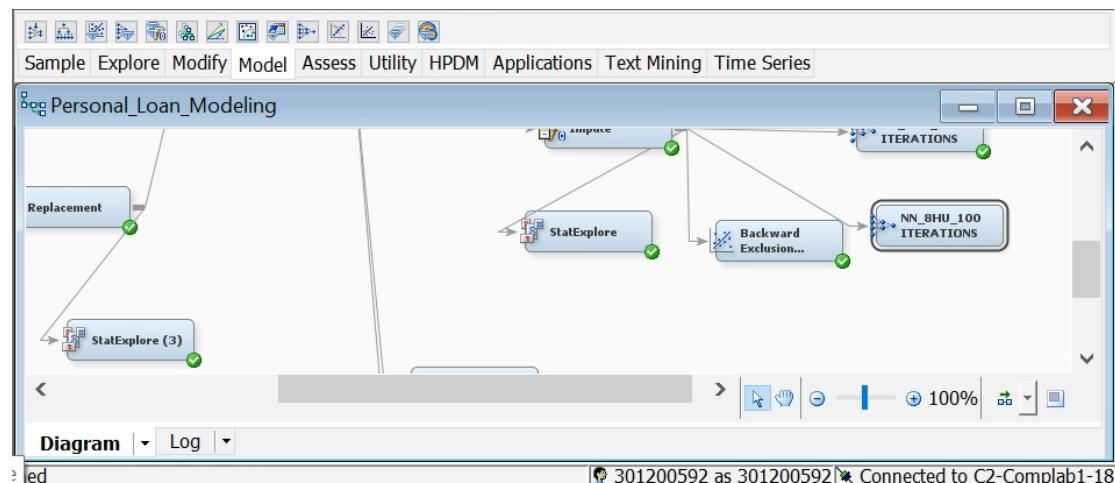


Fig 4.7 NN_8HU_100 Iterations

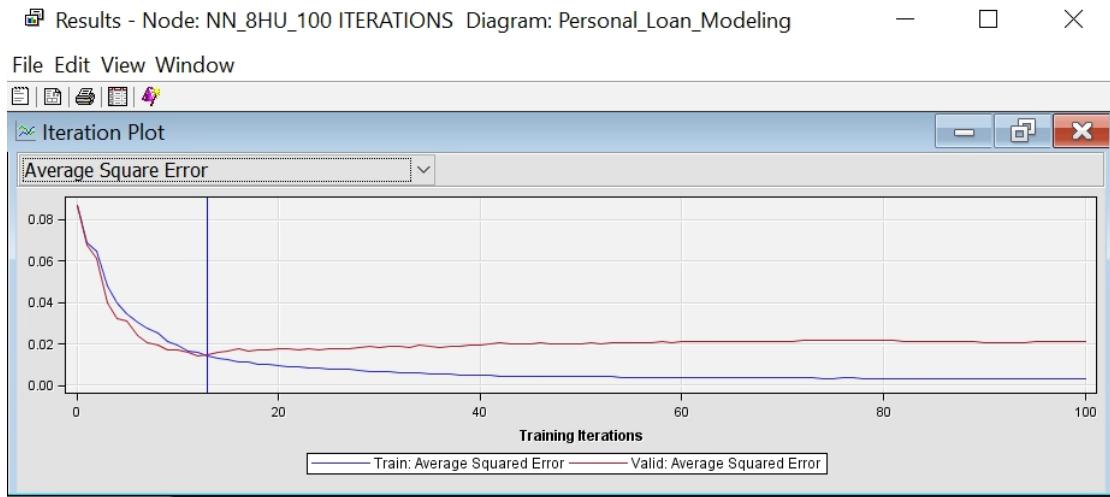


Fig 4.8 Iterative plot of ASE for 8HU_100 Iterations

NEURAL NETWORK ON TRANSFORM NODE (NNT)

As stated earlier, neural network does not do an optimum work with skewness. We minimized our skewed data using our transform node so we will go ahead and check our neural network connected to our transform node.

1) Neural Network of 3HU and 50 Iteration on Transform Node

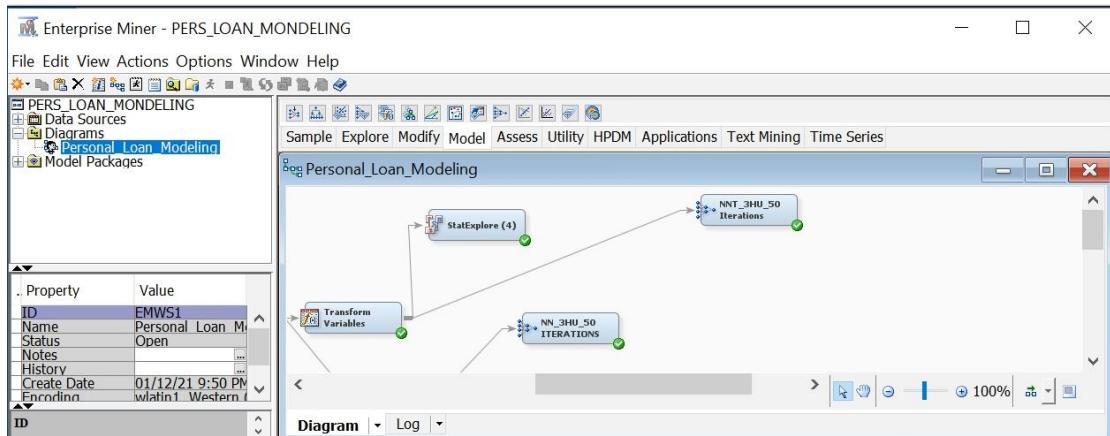


Fig 4.9 NNT of 3HU and 50 Iterations

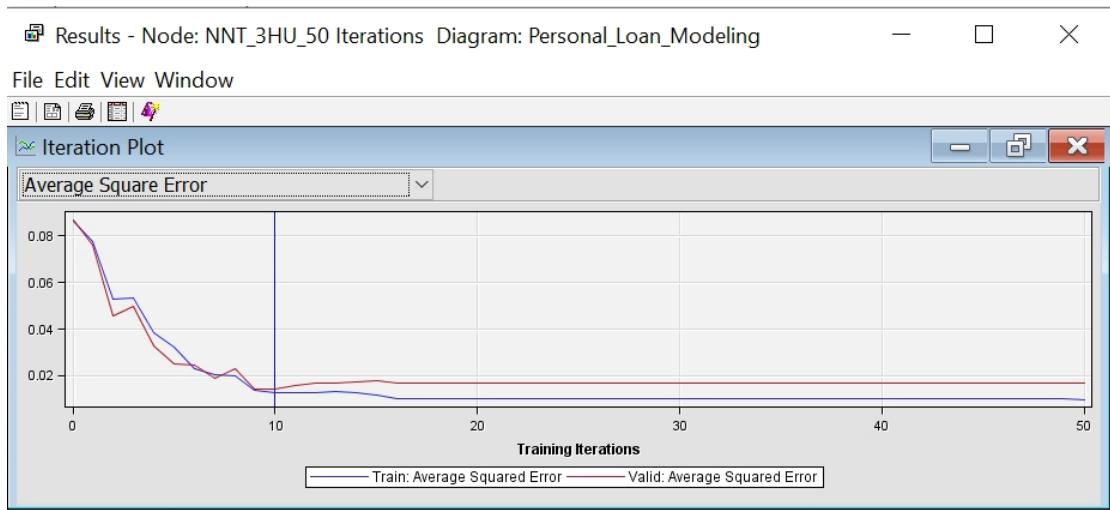


Fig 4.10 Iterative Plot of ASE for NNT_3HU_50 Iterations.

In the same line, we ran neural networks on transform nodes for the different hidden units and iteration as we did using the impute node.

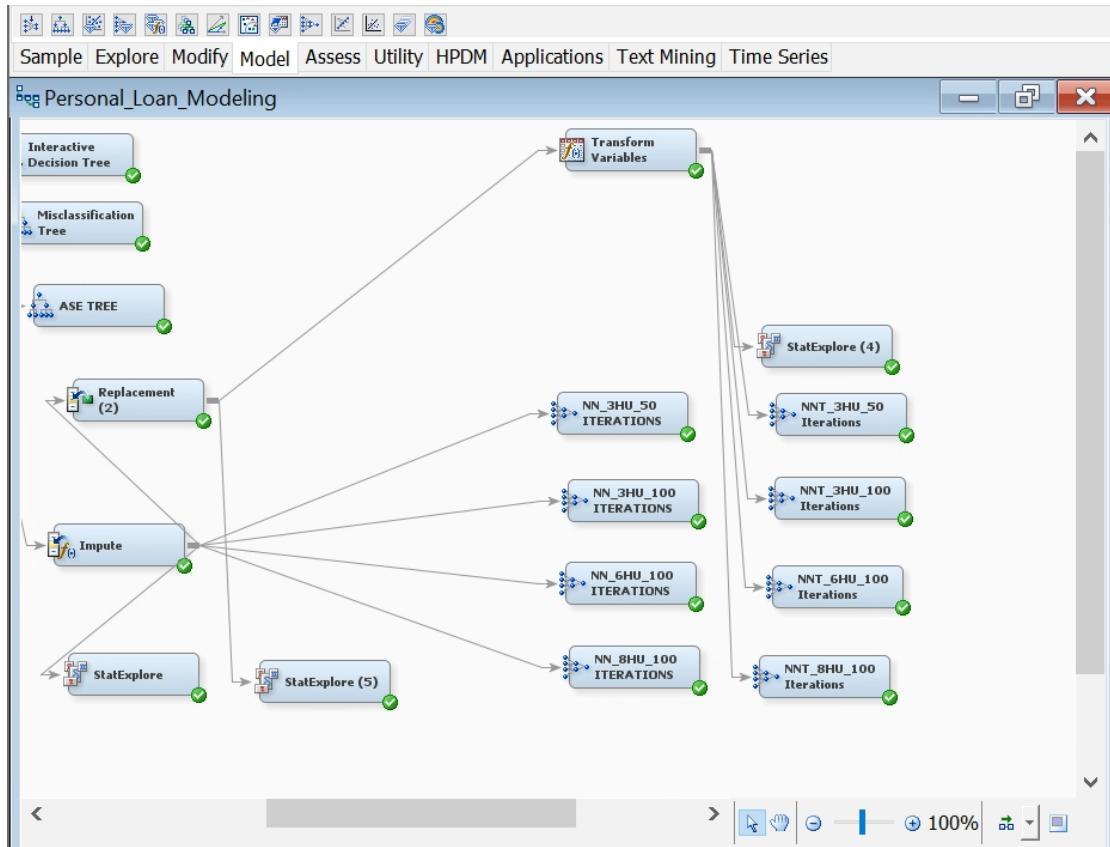


Fig 4.11 Neural Networks of different hidden Units and Iterations on Transform node.

2) Neural Network of 3HU with 100 Iterations

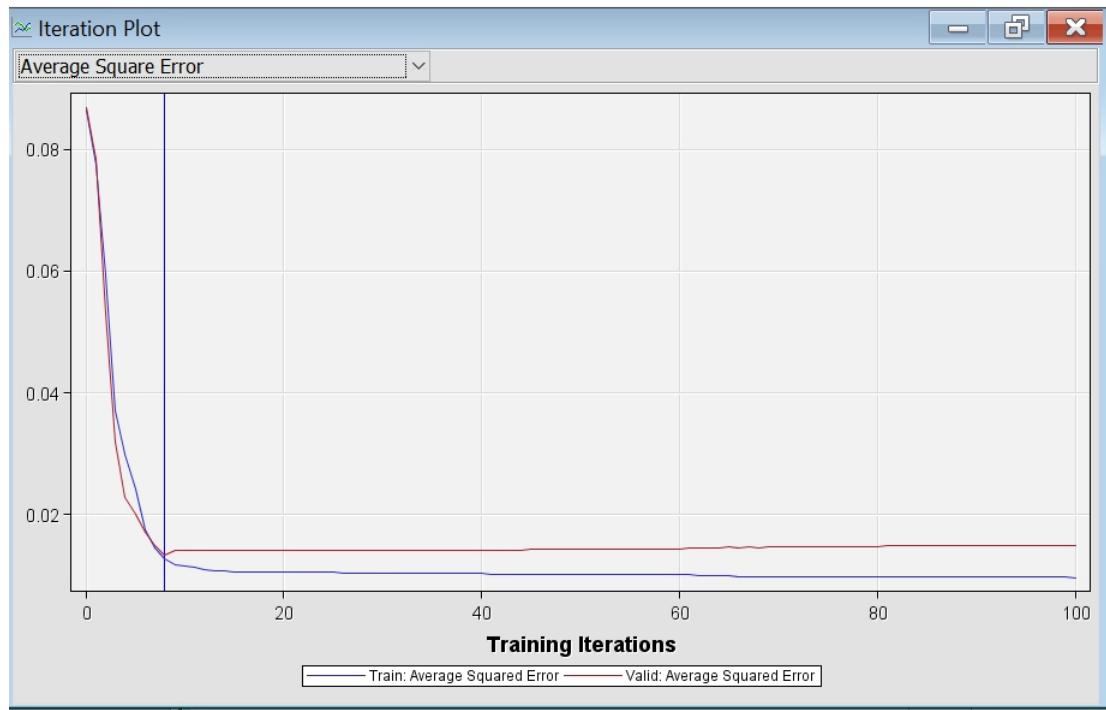


Fig 4.12 Iterative Plot of ASE for NNT_3HU_100 Iterations

3) Neural Network of 6HU with 100 Iterations

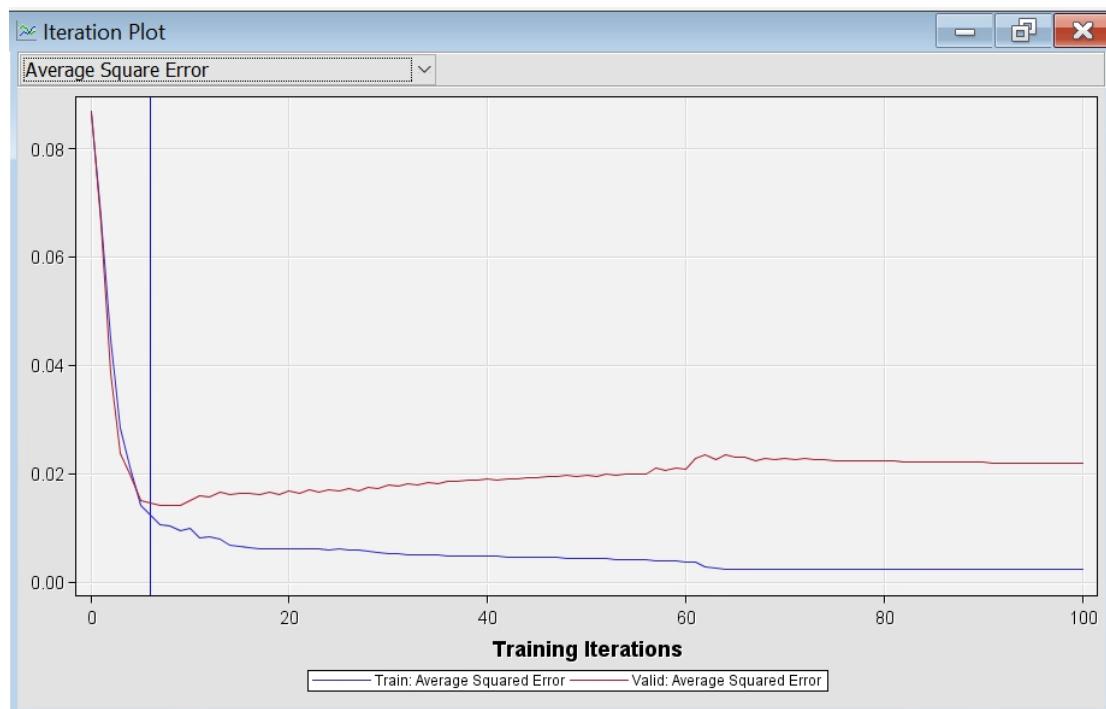


Fig 4.13 Iteration Plot of ASE for NNT_6HU_100 Iterations

4) Neural Network of 8HU with 100 Iterations

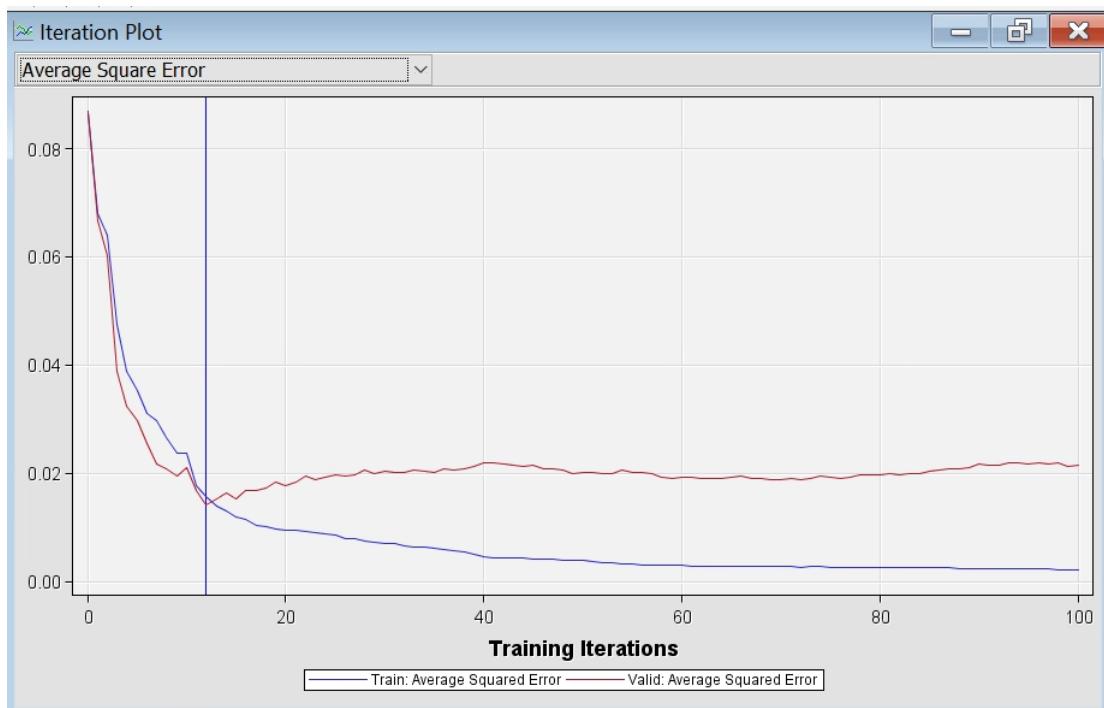


Fig 4.14 Iteration Plot of ASE for NNT_8HU_100_Iterations

Comparison of All Neural Networks based on ASE

Neural Networks	Average Squared Error (ASE)		Misclassification Rate	
	Validation Data	Train Data	Validation Data	Train Data
NN_3HU_50 IT	0.014265	0.011155	0.017186	0.014812
NN_3HU_100 IT	0.014265	0.011155	0.017186	0.014812
NN_6HU_100 IT	0.013468	0.013194	0.016387	0.015612
NN_8HU_100 IT	0.015012	0.014251	0.017586	0.019215
NNT_3HU_50 IT	0.014256	0.012489	0.017586	0.016813
NNT_3HU_100 IT	0.013213	0.012725	0.016787	0.016413

NNT_6HU_100 IT	0.014525	0.012281	0.017586	0.013611
NNT_8HU_100 IT	0.014235	0.015656	0.016387	0.022018

Table 4 Different Neural Networks ASEs

From the table above, our neural network of 6 hidden units and 100 iterations on transform node (NNT_3HU_100 IT) came out as the best model of neural networks based on Average Squared error. Therefore, we would do our model comparison using this neural network.

Neural Network of 3 Hidden Units with 1000 Iterations Built on Backward Regression.

Neural Network is a feed forward type of analysis. Like Regression, they do not work with missing values and their performance increases with better datasets. From our Regression analysis, we deduced that our backward exclusion regression model had the best analysis in terms of average squared error and misclassification rate. Now we will go ahead to feed our dataset from backward regression to our neural network with 3 hidden units and 100 iterations.

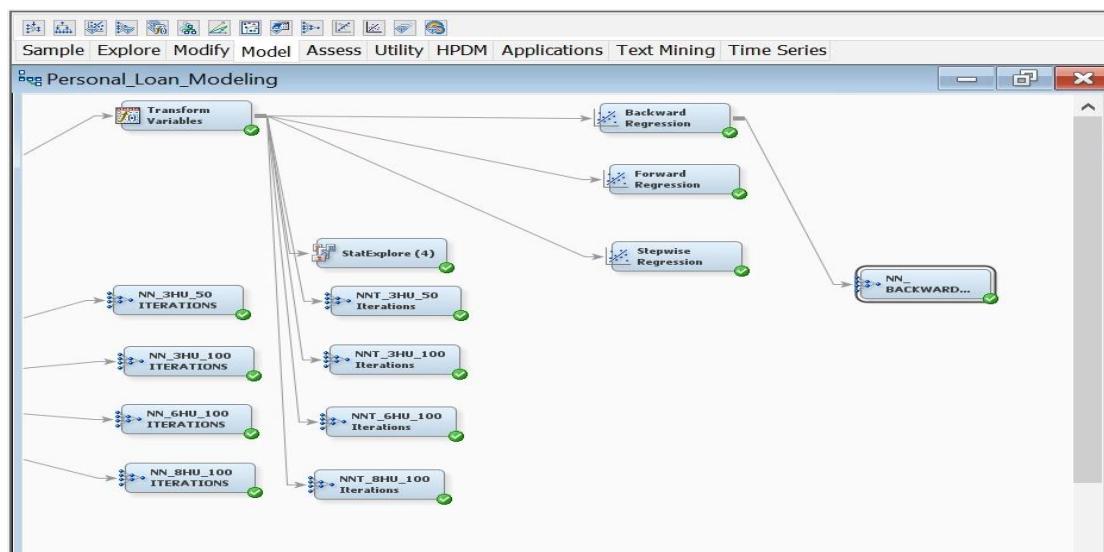


Fig 4.15 NN of 3 Hidden Units and 100 Iterations on Backward Exclusion Regression

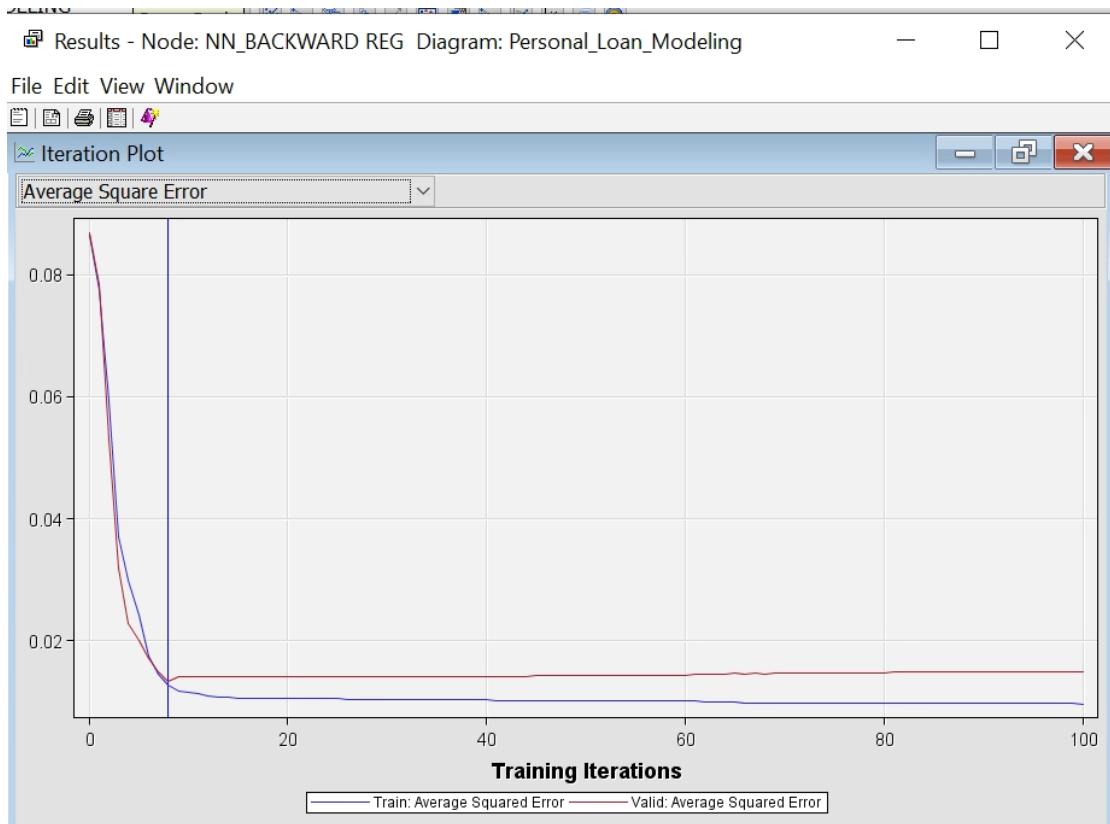


Fig 4.16 Iterative plot of ASE for NN of 3HU_100 Iterations on Backward Regression

As can be seen from the iterative plot of ASE above, our dataset converged on the 8th iteration, though we can see that it started degrading after eight iterations, the degradation of train and validation datasets is very minimal compared to all other neural networks of different iterations and hidden units we have ran so far.

Comparing the two neural networks considered to be the best, we can see below that they produced the same result based on average squared error and misclassification. This is the optimum result of our analysis based on neural network.

	AVERAGE SQUARED ERROR	MISCLASSIFICATION RATE
NNT_3HU_100 IT	0.013213	0.016787
NN_3HU_100 IT ON BACKWARD REGRE	0.013213	0.016787

Table 5 Comparison between NNT_3HU_100 IT and NN_3HU_100 IT on Backward Regression.

CHAPTER 5

MODEL COMPARISM

We have succeeded in running analysis of customer's decision to buy personal loan based on three different predictive models and we have picked the best analysis from each of the three analysis. We will go ahead to compare the bests of these analysis using the model comparison mode, and as usual, based on the average squared error.

We also changed our selection table to run on validation dataset and not train data below.

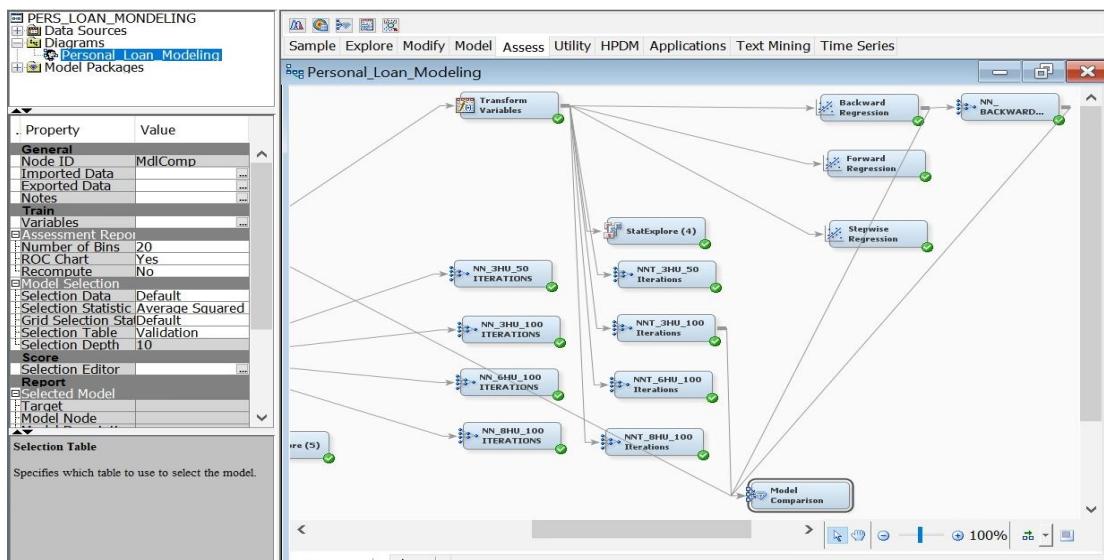


Fig 5.1 MODEL COMPARISON OF ASE DECISION TREE, BACKWARD REGRESSION, NNT_3HU_100 IT AND NN_BACKWARD REGRESSION.

Results - Node: Model Comparison Diagram: Personal_Loan_Modeling

File Edit View Window

The screenshot shows a software interface for model comparison. The title bar reads "Results - Node: Model Comparison Diagram: Personal_Loan_Modeling". The menu bar includes "File", "Edit", "View", and "Window". Below the menu is a toolbar with icons for file operations. The main area is titled "Fit Statistics" and contains a table comparing five different models based on various performance metrics.

ce	Model Node	Model Description	Valid: Average Squared Error	Valid: Roc Index	Target Variable	Target Label	Selection Criterion	Valid: Average Squared Error	Train: Sum of Frequencies	T
16	Neural6	NNT 3HU 100 Itera...	0.013213	0.99	Person...	Person...	0.0132...	2498	0	
19	Neural9	NN BACKWARD R...	0.013213	0.99	Person...	Person...	0.0132...	2498	0	
Tree3	ASE TREE		0.01282	0.995	Person...	Person...	0.01282	2498	0	
Rea	Backward Regression		0.028354	0.971	Person...	Person...	0.0283...	2498	0	

Fig 5.2 Fit Statistic Window for all chosen Models

From the fit statistic window above, our Decision Tree based on Average Squared Error appears to be our best model so far with average square error as 0.01282 and ROC Index of 0.995.

Conclusion and Strategic Report

We will have to give a summary of our prediction or modeling based on our best analysis which is decision tree based on ASE. The summary report will come from the tree we obtained during our analysis using ASE decision tree.

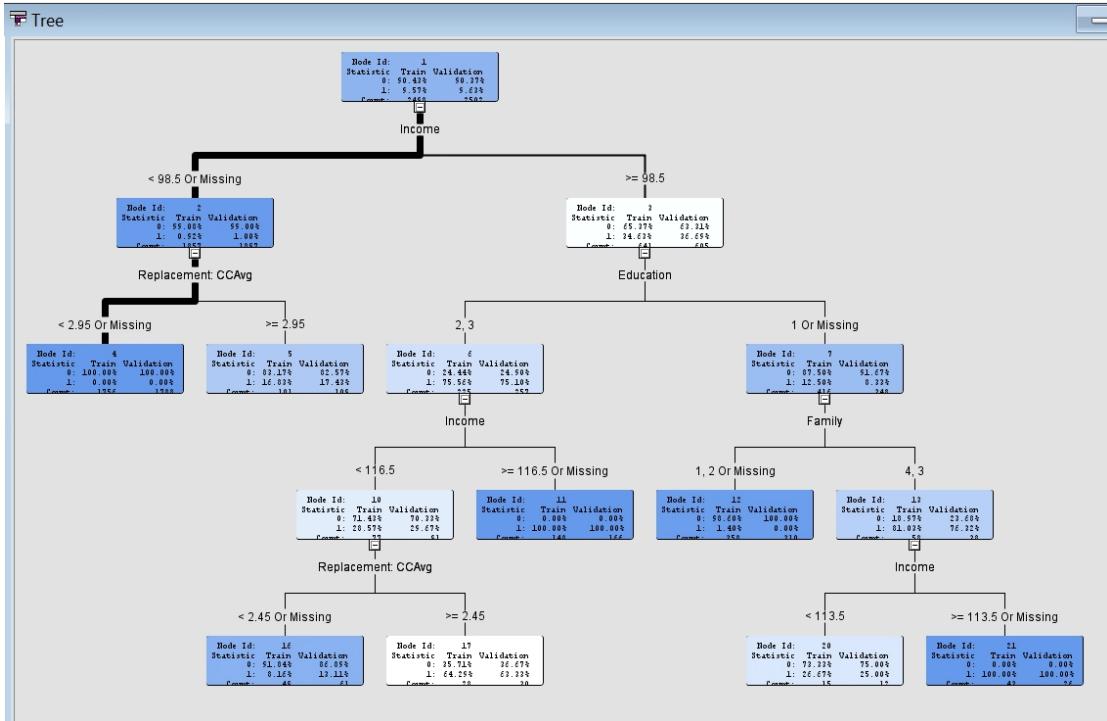


Fig 5.3 Decision Tree Based On ASE

Income seems to be the main determinant in customer's decision to purchase personal loan. Customer's whose incomes are higher or equal to 98.5 thousand US dollars have 36.7% chance of buying personal loan while the ones with lower income have only one per cent chance of buying personal loan. Banks should focus their resources on customers who make more money in a year than those who do not.

Furthermore, under customers with higher income earning is their education level. Education has a whole lot to do with personal loan purchase. Customers who earn higher income and are graduate or have advanced education level stand a 75% chance of buying personal loans compared to those who are students or under graduates in the same income level. Banks should focus on customers who are educationally advanced than those who are not.

To salvage customers who earn higher than 9.8 thousand US dollars with lower education, that is, customers who earn higher income but with lower education or those whose education level indication is missing, banks must focus on those of them with larger family members as they are much more likely to purchase personal loan than their counterparts with fewer family members.

References

Chen, J. (2019). *Neural Network Definition*. Investopedia.

<https://www.investopedia.com/terms/n/neuralnetwork.asp>

Data Mining Software, Model Development and Deployment, SAS Enterprise Miner.

(n.d.). https://www.sas.com/en_ca/software/enterprise-miner.html

Decision Trees. (n.d.). Scikit-Learn. Retrieved December 17, 2021, from <http://scikit-learn.org/stable/modules/tree.html>

Logit Regression / SAS Data Analysis Examples. (n.d.). Stats.idre.ucla.edu.

<https://stats.idre.ucla.edu/sas/dae/logit-regression/>

Predictive Analytics: What it is and why it matters. (n.d.). [ww.sas.com](https://www.sas.com/en_in/insights/analytics/predictive-analytics.html).

https://www.sas.com/en_in/insights/analytics/predictive-analytics.html

What is Linear Regression? (n.d.). Statistics Solutions.

<https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/what-is-linear-regression/>

Wikipedia Contributors. (2019, April 5). *Artificial neural network*. Wikipedia;

Wikimedia Foundation. https://en.wikipedia.org/wiki/Neural_Network