Lending Club Loan Data

An analysis using SAS Enterprise Miner

SUBMITTED BY: GROUP 02

Gopesh Vijayvergiya
Rasika Kulkarni
Kriti Jain
Rajnath Geddi

INDEX

1. EXECUTIVE SUMMARY

1.1 Mission Statement

The objective of this project to build two models:

- a model that will predict if a loan will be fully funded by investors or not
- a model that predicts if a loan will be defaulted or not

1.2 Introduction

Lending Club (LC) is an online peer to peer lending platform headquartered in San Francisco, California. It facilitates investors in searching and browsing the loan listings and helps them select loans that they want to invest in. Investors make money from interest. Lending Club makes money by charging borrowers an origination fee and investors a service fee.

For assessing the risk associated with their borrowers, Lending Club primarily relies on a grade and sub-grade system that it assigns them based on their credit history. This information is then made available to investors who fund the loan requests, so that the investors can decide which loan request and how much of that loan request they will fund. In addition to the grade information, Lending Club provides historical loan performance data to investors for more comprehensive analysis.

2. PROJECT MOTIVATION

We selected this dataset because our team was unfamiliar with the domain and it would add the challenge of learning about the data while trying to extract meaningful information from of it. The dataset had many observations and attributes and it would give us the opportunity to explore different techniques in data pre-processing and data mining.

3. DATA DESCRIPTION

The data is second-hand data that was obtained from www.kaggle.com. The dataset contains the data of only approved loans by the LendingClub between 2007 and 2015.

The dataset has 75 variables and 880,000+ observations. The attributes include information about the borrowers such as their open accounts, months since last delinquency, amount of loan requested, amount of loan funded by investor, employment length, annual income, lending club assigned loan grade, interest rate, installments etc.

Following are our target variables:

- not_fully_funded (binary - 0 - No / 1 - Yes)
- will_default (binary - 0 - No / 1 - Yes)

We have rejected variables having missing values greater than 50% and those that were not relevant for the analysis such as

- IDs
- URL
- Employee Title
- Description

4. BI Model: Target variable - Not_fully_funded

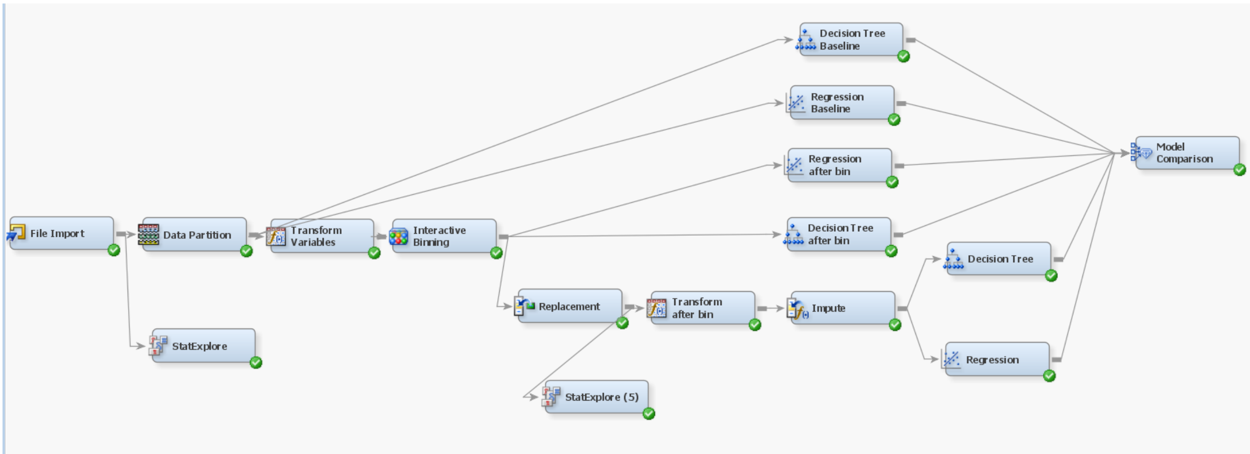


Fig: 4.1 – Process flow diagram for target variable not_fully_funded

We have created the target variable not_fully_funded that represents whether the loan is fully funded by an investor or not.

4.1 DATA EXPLORATION

The initial exploration was done using the File Import node by turning on the summarize option to 'Yes'. We can see from the below statistics table variables with high missing values:

Statistics Table

Variable Name	Type	Number of Levels	Percent Missing ▼
dti joint	N	.	99.94264
verification status joint	C	3	99.94241
annual inc joint	N	.	99.94241
il util	N	.	97.90202
mths since rcnt il	N	.	97.65489
all util	N	.	97.59156
inq fi	N	.	97.59156
inq last 12m	N	.	97.59156
max bal bc	N	.	97.59156
open acc 6m	N	.	97.59156
open il 12m	N	.	97.59156
open il 24m	N	.	97.59156
open il 6m	N	.	97.59156
open rv 12m	N	.	97.59156
open rv 24m	N	.	97.59156
total bal il	N	.	97.59156
total cu tl	N	.	97.59156
mths since last record	N	.	84.5553
mths since last major derog	N	.	75.01597
mths since last delinq	N	.	51.19706
next pymnt d	N	.	28.50766

Fig: 4.2 – List of missing values

The StatExplorer node ranks all the variables based on their worth in predicting the target variable. We also need to identify the variables with high worth having large number of missing values and need to reject or impute them.

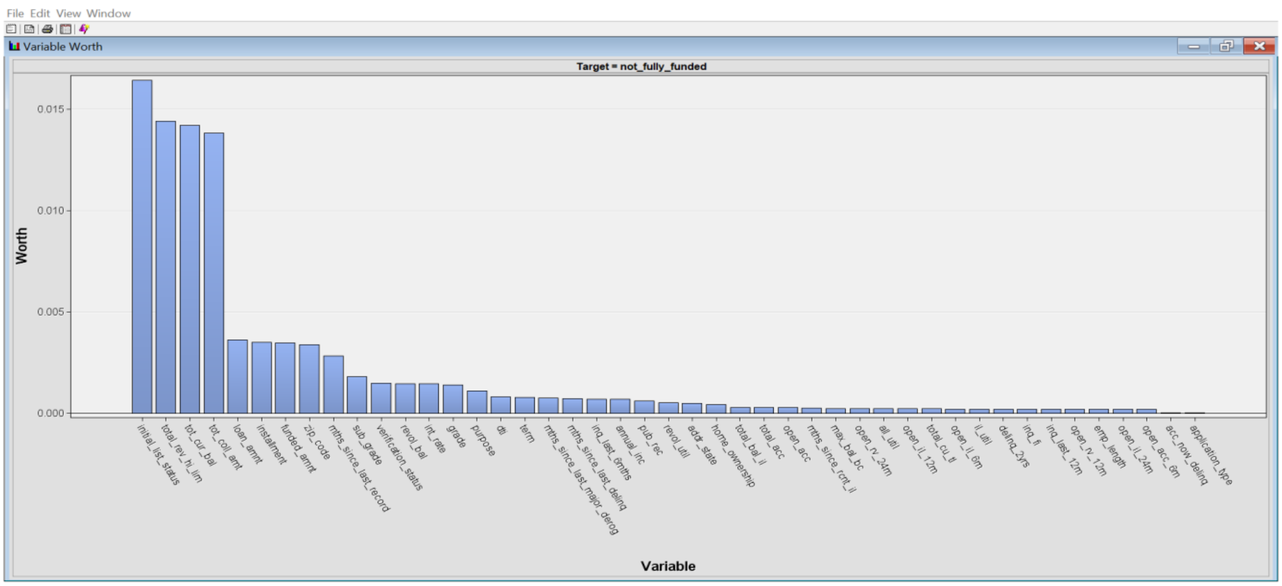


Fig: 4.3 – Variable worth for the target variable not_fully_funded

Target variable breakdown:

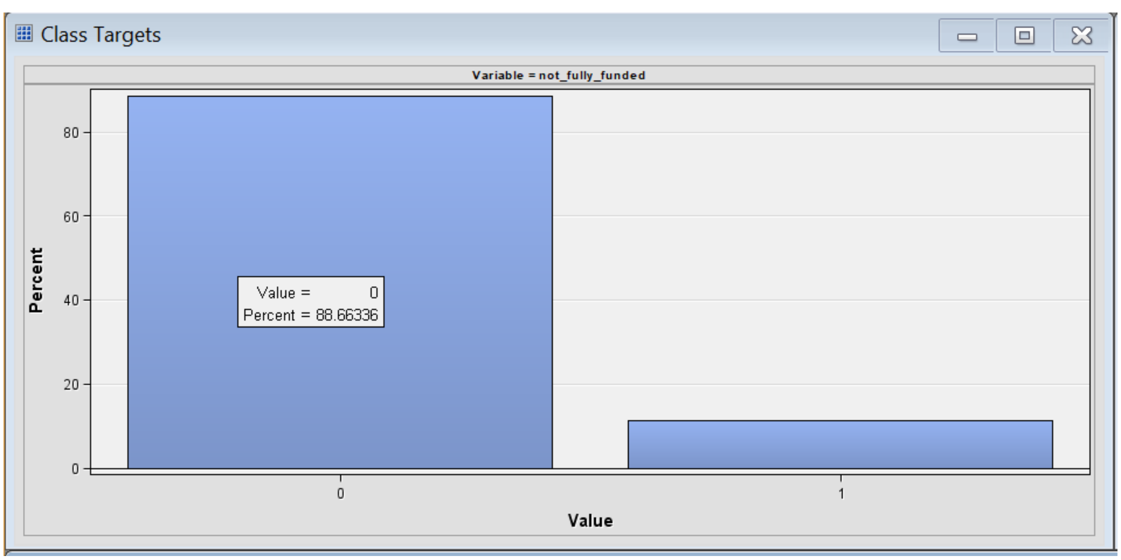


Fig: 4.4 – Classification of the target variable not_fully_funded

From the fig 4.4, we can see that the baseline misclassification rate is **11.33%**

4.2 DATA PARTITIONING

We have partitioned the data into training as 60% and validation as 40% of the initial dataset. The training data is used to create models to predict the outcome while the validation data helps to assess the model.

4.3 DATA PRE-PROCESSING

Dropping variables: Since we created the target variable Not_fully_funded from the two variables funded_amt and funded_amt_inv, we have rejected funded_amt_inv as incorporating it in the analysis was resulting in near perfect model due to its direct relation with the target variable.

Transform variables: We have created 3 variables open_rv, total_open_il, Total_open_acc by combining already existing columns in our dataset to simplify the computation. open_rv was created combining open_rv_12m and open_rv_24m i.e. number of revolving trades opened in past 12 and 24 months. We did the same thing for total_open_il by combining open_il_6m, open_il_12m, open_il_24m i.e. the number of installment accounts opened in the past 6,12 and 24 months. And lastly, we created Total_open_acc by combining open_acc and open_acc_6m which represents the number of open credit lines in the borrower's credit files.

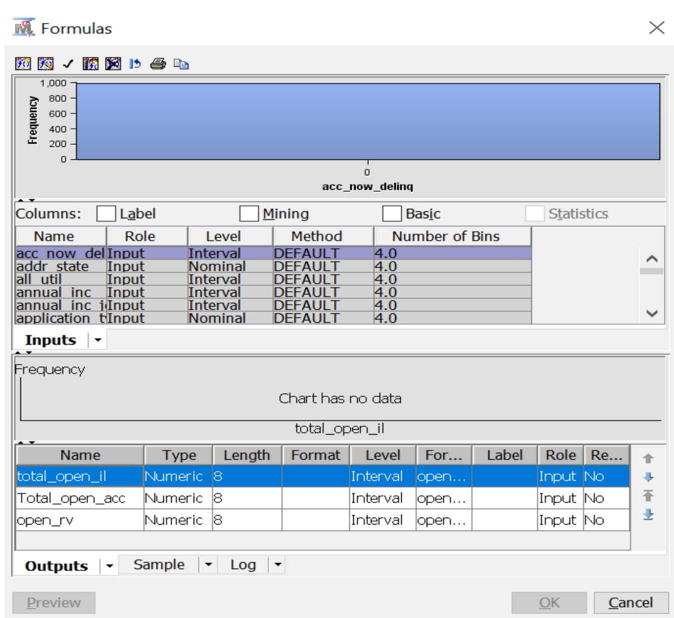


Fig: 4.5 – Creation of three new variables by combining existing variables

4.4 INTERACTIVE BINNING

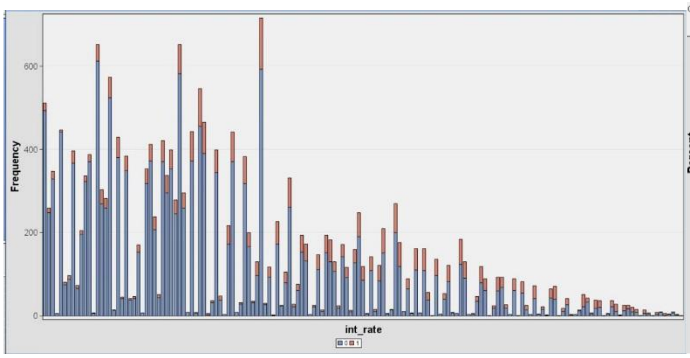
Transform Node: We used transform node to transform variables such as:

- Months since last delinquency
- Months since last public record
- Months since last derogatory comment

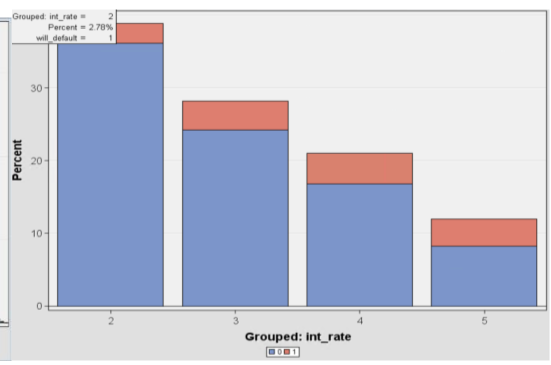
These variables had approximately 97% missing values, but we have assumed that a missing value means that it has been a very long time since or the person has never had a record against them. For this reason, we have transformed these variables using transform and interactive binning to be used has whether a person has a record against them (irrespective of when) or not.

Conversion of continuous variables to categorical:

To get more discernible results, we used interactive binning to convert continuous variables like interest rate, dti, last payment amount, total payment, employment length, delinquencies in the past 2 years into categorical variables.



Before Categorizing



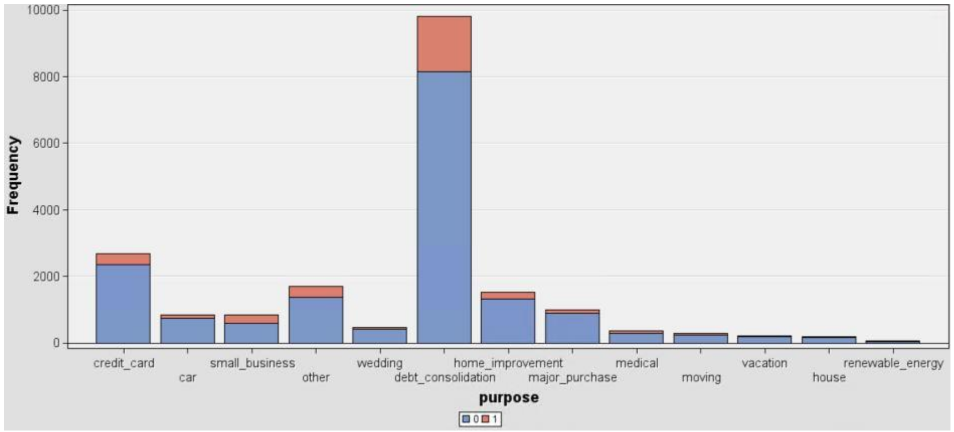
After Categorizing

Fig: 4.6 – Categorizing continuous variable int_rate

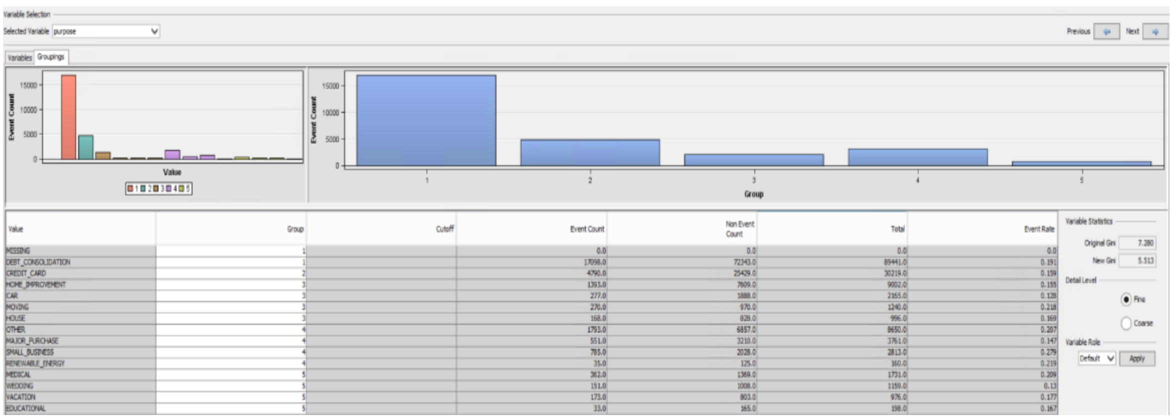
Fig: 4.6 shows that as the interest rate increases, the percent of defaulted loans (1) also increases.

Category reduction of nominal variable - purpose:

The nominal variable purpose had 14 levels, we created 5 categories by combining similar purposes.



Before categorizing



Value	Group	Cutoff	Event Count	Non Event Count	Total	Event Rate
MISSING	1		0.0	0.0	0.0	0.0
DEBT_CONSOLIDATION	1		17098.0	72343.0	89441.0	0.191
CREDIT_CARD	2		4790.0	25429.0	30219.0	0.159
HOME_IMPROVEMENT	3		1393.0	7609.0	9002.0	0.155
CAR	3		277.0	1888.0	2165.0	0.128
MOVING	3		270.0	970.0	1240.0	0.218
HOUSE	3		168.0	828.0	996.0	0.169
OTHER	4		1793.0	6857.0	8650.0	0.207
MAJOR_PURCHASE	4		551.0	3210.0	3761.0	0.147
SMALL_BUSINESS	4		785.0	2028.0	2813.0	0.279
RENEWABLE_ENERGY	4		35.0	125.0	160.0	0.219
MEDICAL	5		362.0	1369.0	1731.0	0.209
WEDDING	5		151.0	1008.0	1159.0	0.13
VACATION	5		173.0	803.0	976.0	0.177
EDUCATIONAL	5		33.0	165.0	198.0	0.167

After categorizing

Fig: 4.7 – Reducing categories of nominal variable purpose

Removing outliers and reducing the skewness: The replacement, transform and impute nodes were used in conjunction for removing the outliers and reducing the skewness of the data.

Replacement node: The replacement node was used to remove values that were outside of +/- 3 standard deviations away from the mean.

Transform Node: We have used the transform node to reduce the skewness of variables using a logarithmic transformation for all the variables. This helped us in drastically reducing the skewness of highly skewed variables.

Impute: We have used the default impute method which uses the mean to impute the missing values for interval variables and count to impute for class variables.

Following are the results after removing outliers, transforming and imputing:

Imputation Summary

Variable Name	Impute Method	Imputed Variable	Impute Value	Role	Measurement Level	Label	Number of Missing for TRAIN
REP annual inc	MEAN	IMP REP annual inc	73640.79	INPUT	INTERVAL	Replacement: annual inc	2
REP inq last 6mths	MEAN	IMP REP inq last 6mths	0.680345	INPUT	INTERVAL	Replacement: inq last 6mt...	13
REP open acc	MEAN	IMP REP open acc	11.48789	INPUT	INTERVAL	Replacement: open acc	13
REP pub rec	MEAN	IMP REP pub rec	0.177163	INPUT	INTERVAL	Replacement: pub rec	13
REP revol util	MEAN	IMP REP revol util	55.06145	INPUT	INTERVAL	Replacement: revol util	303
REP tot coll amt	MEAN	IMP REP tot coll amt	204.8739	INPUT	INTERVAL	Replacement: tot coll amt	41921
REP tot cur bal	MEAN	IMP REP tot cur bal	136369.9	INPUT	INTERVAL	Replacement: tot cur bal	41921
REP total acc	MEAN	IMP REP total acc	25.20061	INPUT	INTERVAL	Replacement: total acc	13
REP total rev hi lim	MEAN	IMP REP total rev hi lim	31152.08	INPUT	INTERVAL	Replacement: total rev hi ...	41921

Fig: 4.8 – Results – removing outliers, reducing skewness and imputing

4.5 Models for target variable not_fully_funded

We used decision tree and logistics regression and tried 6 different models.

4.6 Model Comparison Results

Below are the results of the model comparison node. Using misclassification rate as the selection criterion, SAS EM has selected "**Tree7 (Decision tree after bin)**" as the best model.

Fit Statistics

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Selection Criterion : Valid: Misclassification Rate	Valid: Average Squared Error	Valid: Mean Square Error	Valid: Root Mean Square Error	Train: Misclassification Rate	Train: Average Squared Error	Train: Root Mean Squared Error
	Reg3	Reg3	Regression Baseline	not fully funded	0.11337	0.11337	0.11337	0.336705	0.113364	0.113364	0.336696
	Reg7	Reg7	Regression after bin	not fully funded	0.11339	0.100886	0.100886	0.317625	0.113342	0.100836	0.317573
	Tree3	Tree3	Decision Tree Baseline	not fully funded	0.1078...	0.086674	.	.	0.107506	0.08662	.
	Reg8	Reg8	Regression	not fully funded	0.1130...	0.086023	0.086023	0.293297	0.112662	0.085613	0.292886
	Tree8	Tree8	Decision Tree	not fully funded	0.10556	0.082651	.	.	0.104129	0.081827	.
Y	Tree7	Tree7	Decision Tree after bin	not fully funded	0.1052	0.081541	.	.	0.104738	0.081136	.

Fig: 4.9 – Fit statistics of models for target variable not_fully_funded

Below are the ROC curves for different models that we tried. The ROC curves show that the "**Tree7 (Decision tree after bin)**" is the best model.

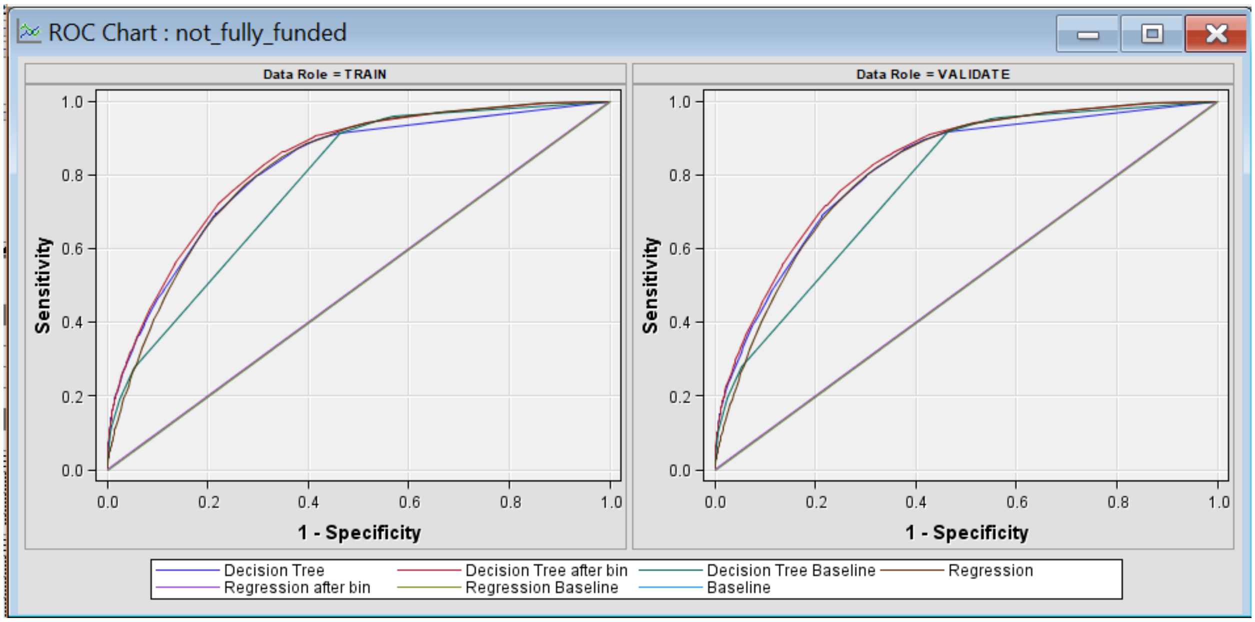


Fig: 4.10 – ROC curves of models for target variable not_fully_funded

Below is the cumulative lift chart for all the models:

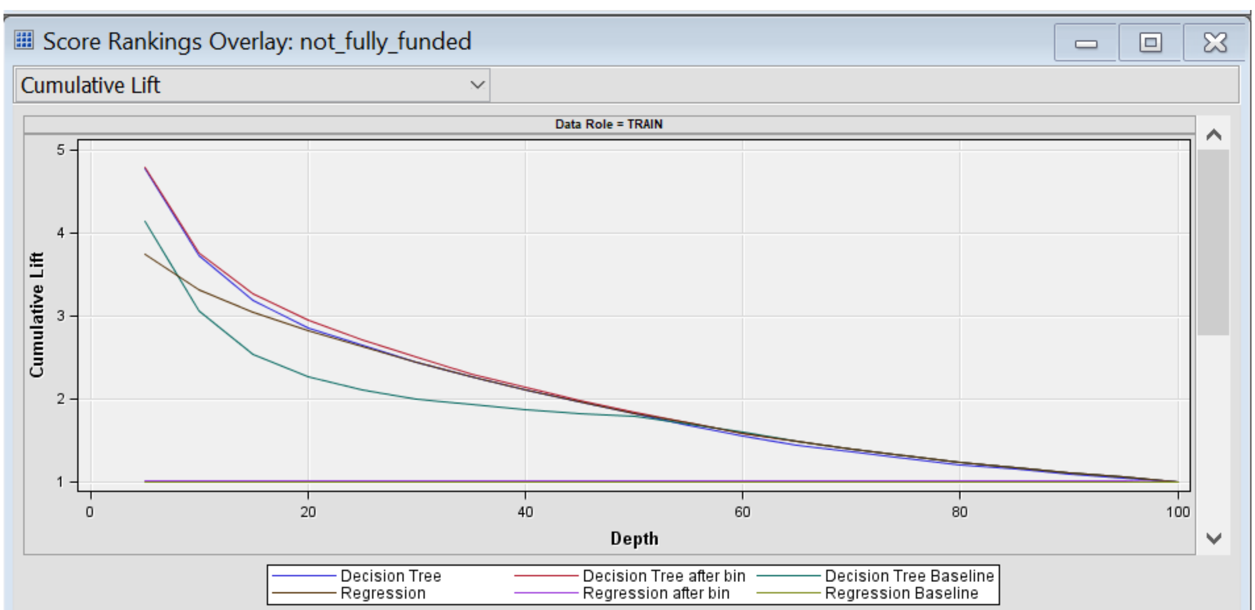


Fig: 4.11 – Cumulative lift of models for target variable not_fully_funded

Below are the results for our best model:

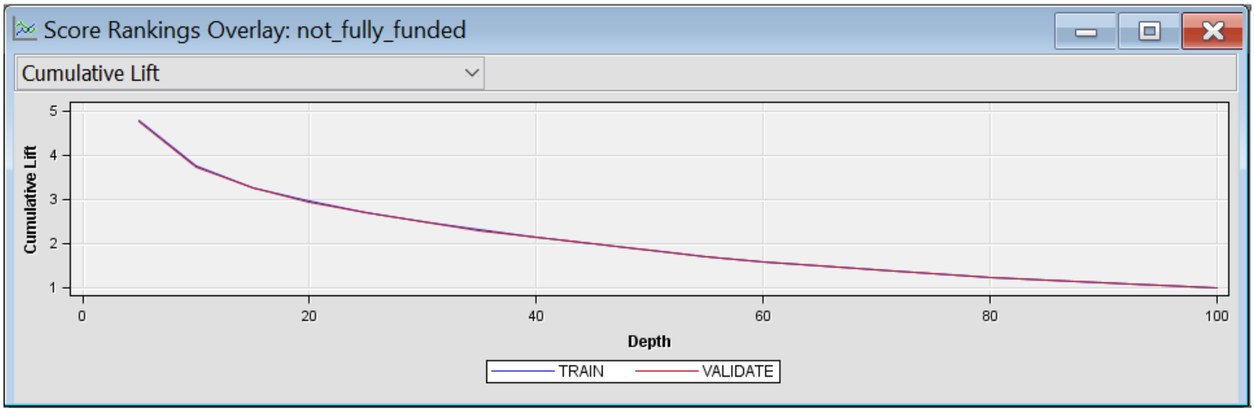


Fig: 4.12 – Cumulative lift of Tree7 for target variable not_fully_funded

From the Fig: 4.12, we can see that for 40% of the total approved loans, 2.11 times the loan will not be fully funded by the investors.

Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
not fully funded	not fully funded	NOBS	Sum of Freque...	532426	354953	.
not fully funded	not fully funded	MISC	Misclassificati...	0.104738	0.1052	.
not fully funded	not fully funded	MAX	Maximum Abs...	0.995535	1	.
not fully funded	not fully funded	SSE	Sum of Squar...	86397.85	57886.32	.
not fully funded	not fully funded	ASE	Average Squa...	0.081136	0.081541	.
not fully funded	not fully funded	RASE	Root Average ...	0.284844	0.285554	.
not fully funded	not fully funded	DIV	Divisor for ASE	1064852	709906	.
not fully funded	not fully funded	DFT	Total Degrees...	532426	.	.

Fig: 4.13 – Fit statistics for Tree7 for target variable not_fully_funded

Confusion matrix:

		Actual	
		0	**1**
Predicted	**0**	311999 (True -)	34628 (False -)
	1	2713 (False +)	5613 (True +)

Fig: 4.14 – Confusion matrix for Tree7 for target variable not_fully_funded

```
Data Role=TRAIN Target=not_fully_funded Target Label=not_fully_funded

  False       True        False       True
Negative    Negative    Positive    Positive

  51880      468183       3885        8478


Data Role=VALIDATE Target=not_fully_funded Target Label=not_fully_funded

  False       True        False       True
Negative    Negative    Positive    Positive

  34628      311999       2713        5613
```

Fig: 4.15 – Event classification table for Tree7 for target variable not_fully_funded

4.7 Business implications

It is important to estimate the effect of misclassify the target variable. The errors can be classified into two types:

- False Positive
- False Negative

The more expensive error is the false negative error where the model predicts a 0 - meaning a loan is fully funded when it is actually not i.e. 1. The implications for this is that the lending club may not hold sufficient reserve funds to fund the loan thereby losing out on profit and customers.

4.8 Conclusion

This model was designed to help Lending club decide how much money they should keep in reserve, if they were to fund loans that were not fully funded by investors. Our best model returned a misclassification rate of 10.52% and we were unable to reduce it further. We suspect that the data is insufficient in the respect that we don't have the information about the credit history and credit score of the borrowers.

5. BI Model: Target Variable - Will Default

The following model will be used to predict the outcome of a loan whether it will be defaulted or not:

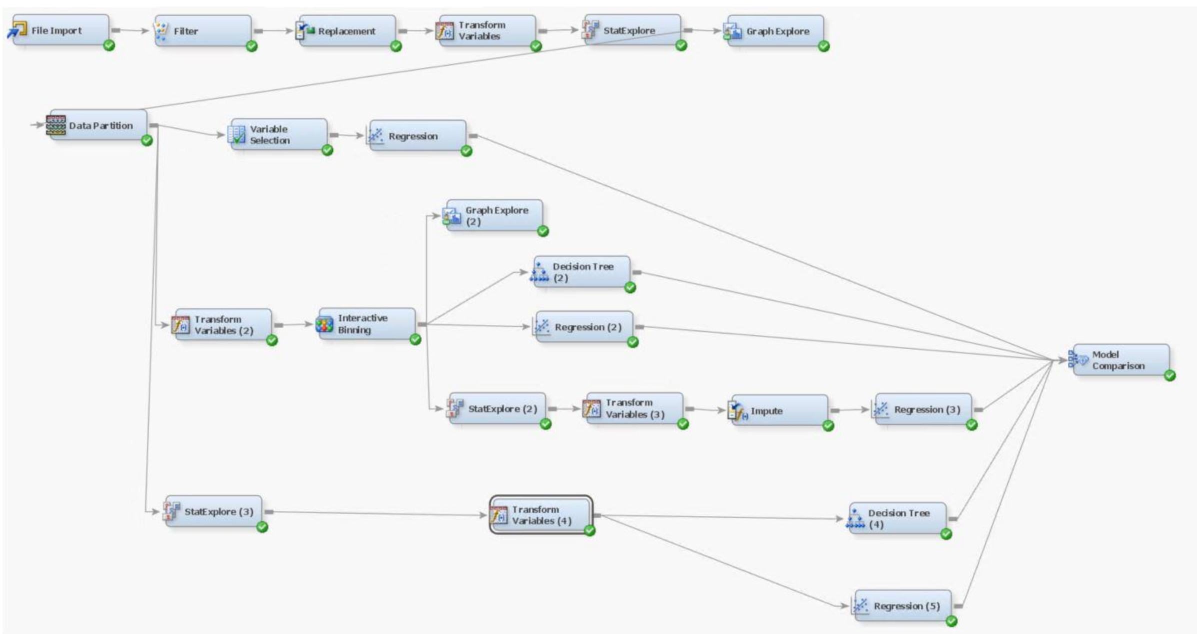


Fig: 5.1 – Process flow diagram for target variable will_default

5.1 DATA EXPLORATION

From the diagram below, we can see that the baseline misclassification error is 19.4%.

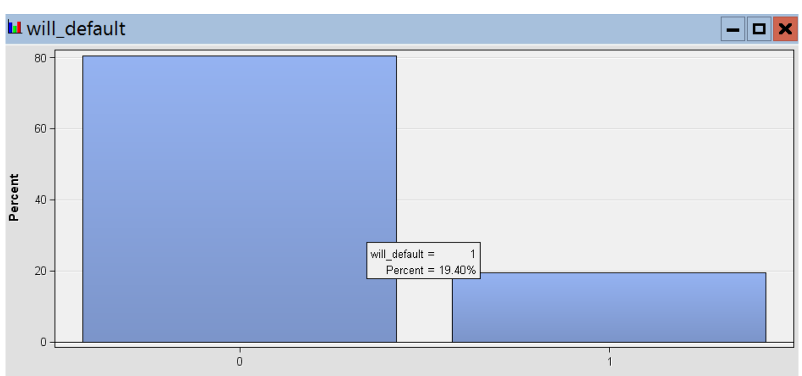


Fig: 5.2 – Classification for target variable will_default

5.2 DATA PRE-PROCESSING

We have used a filter to remove observations that have a loan status of current, in grace period, late, and issued. These observations were removed since our target is to predict if a loan will be defaulted on or not. We were left with 250,000+ observations after filtering out the classes of target variables that were unnecessary to our model.

Target Variable: Replacement and Transform nodes were used to create a new binary target variable – Will_Default where 0 – No (Loan Fully Repaid), 1 – Yes (Loan will be defaulted on).

Following is the picture showing the transformation for the target variable:

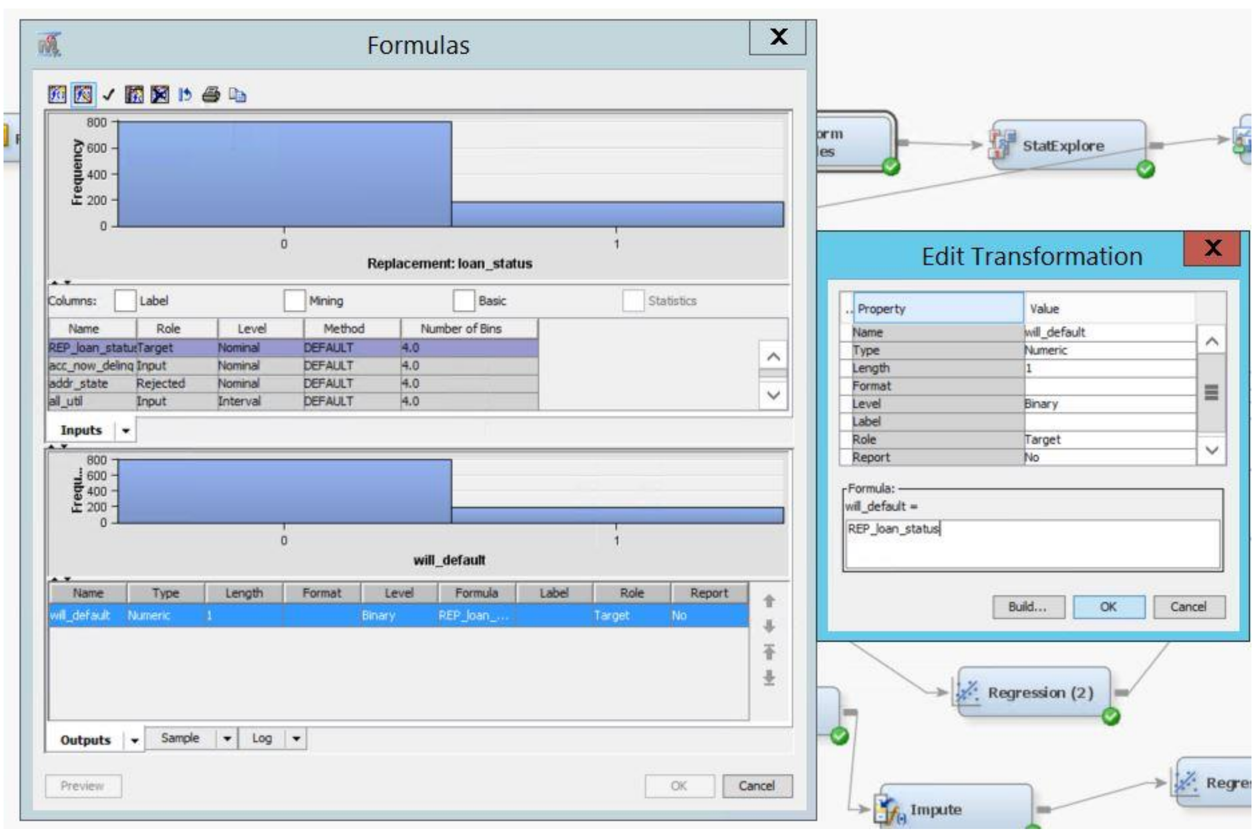


Fig: 5.3 – Creating target variable will_default

Data Partitioning: The data was partitioned as Train – 60% and Validate – 40%.

Dropping variables: We rejected variables that had missing values above 50% and very less variable worth. We rejected variables like recoveries, coll_rec_fee (collection recovery fee), out_prncp (outstanding principal) as incorporating it in the analysis was resulting in near perfect model due to its direct relation with the target variable. We also rejected variables like last_pymnt (last payment), tot_pymnt (total payment), tot_received_prncp (total received principal) and tot_received_int (total received interest) as this information would be unavailable at the time of deciding whether to fund a loan or not.

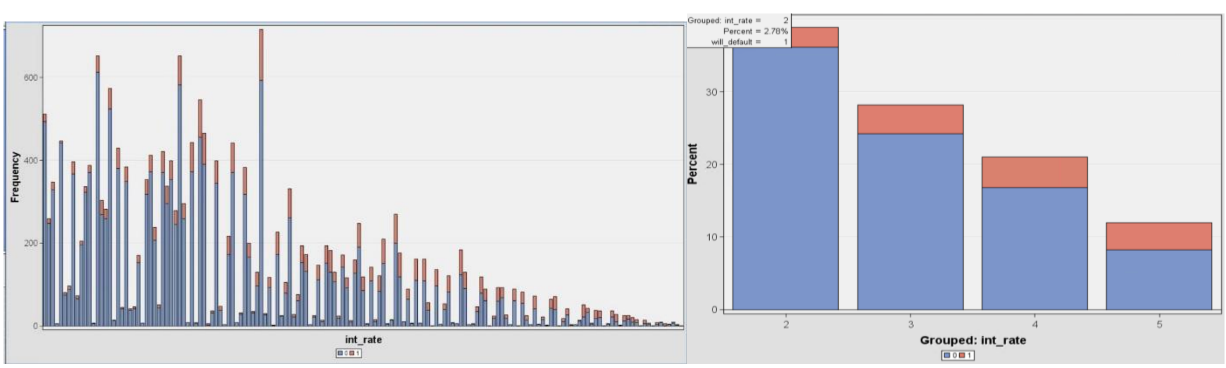
Transform Node: We used transform node to transform variables such as:

- Months since last delinquency
- Months since last public record
- Months since last derogatory comment

These variables had approximately 97% missing values, but we have assumed that a missing value means that it has been a very long time since or the person has never had a record against them. For this reason, we have transformed these variables using transform and interactive binning to be used has whether a person has a record against them (irrespective of when) or not.

Conversion of continuous variables to categorical:

To get more discernible results, we used interactive binning to convert continuous variables like interest rate, dti, last payment amount, total payment, employment length, delinquencies in the past 2 years into categorical variables.



Before Categorizing

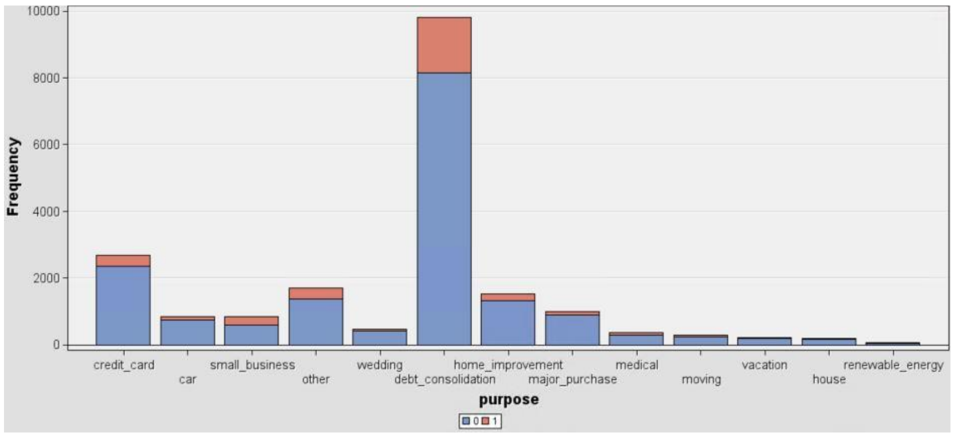
After Categorizing

Fig: 5.4 – Categorizing continuous variable int_rate

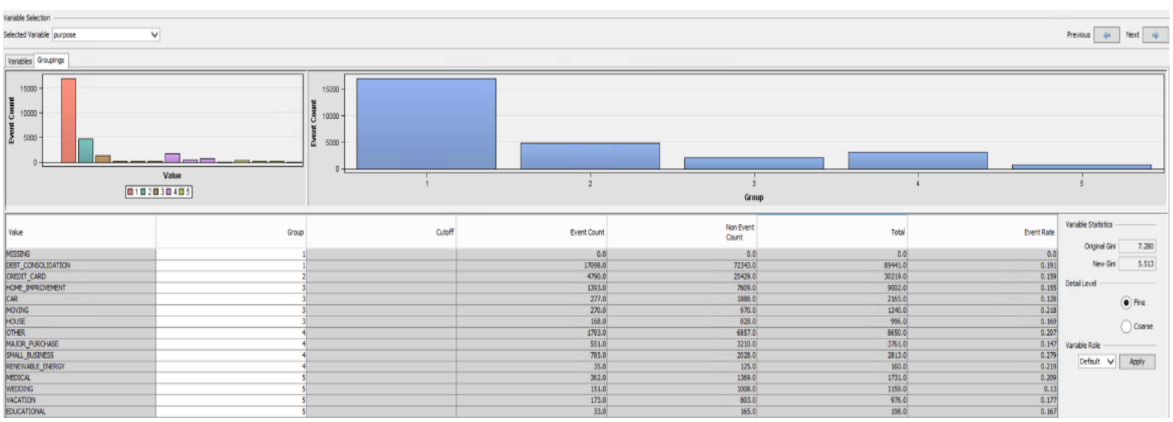
Fig: 5.4 shows that as the interest rate increases, the percent of defaulted loans (1) also increases.

Category reduction of nominal variable - purpose:

The nominal variable purpose had 14 levels, we created 5 categories by combining similar purposes.



Before categorizing



Value	Group	Cutoff	Event Count	Non Event Count	Total	Event Rate
MISSING	1		0.0	0.0	0.0	0.0
DEBT_CONSOLIDATION	1		17098.0	72343.0	89441.0	0.191
CREDIT_CARD	2		4790.0	25429.0	30219.0	0.159
HOME_IMPROVEMENT	3		1393.0	7609.0	9002.0	0.155
CAR	3		277.0	1888.0	2165.0	0.128
MOVING	3		270.0	970.0	1240.0	0.218
HOUSE	3		168.0	828.0	996.0	0.169
OTHER	4		1793.0	6857.0	8650.0	0.207
MAJOR_PURCHASE	4		551.0	3210.0	3761.0	0.147
SMALL_BUSINESS	4		785.0	2028.0	2813.0	0.279
RENEWABLE_ENERGY	4		35.0	125.0	160.0	0.219
MEDICAL	5		362.0	1369.0	1731.0	0.209
WEDDING	5		151.0	1008.0	1159.0	0.13
VACATION	5		173.0	803.0	976.0	0.177
EDUCATIONAL	5		33.0	165.0	198.0	0.167

After categorizing

Fig: 5.5 – Reducing categories of nominal variable purpose

To reduce the skewness of the following variables we used the Log 10 method in the Transform node:

Transformations Statistics				
Source ▲	Method	Variable Name	Formula	Skewness
Input	Original	annual_inc		27.8173
Input	Original	dti		0.213386
Input	Original	revol_bal		12.86989
Input	Original	tot_coll_amt		336.3033
Input	Original	total_rev_hi_lim		7.793197
Output	Computed	LG10_annual_inc	log10(annual_inc ...	0.173422
Output	Computed	LG10_dti	log10(dti + 1)	-1.29266
Output	Computed	LG10_revol_bal	log10(revol_bal +...	-3.08175
Output	Computed	LG10_tot_coll_amt	log10(tot_coll_amt...	2.714481
Output	Computed	LG10_total_rev_h...	log10(total_rev_hi...	-1.02543

Fig: 5.6 – Transformation to reduce skewness

Two variables had missing values of 1% and 8% and they were imputed with the median value (replacing by mean did not change the misclassification rate of the model).

Variable Name	Impute Method	Impute Value
revol_util	MEDIAN	55.8
tot_cur_bal	MEDIAN	80124

Fig: 5.7 – Imputing the missing values

Variable selection using Variable Selection Node:

We used the variable selection node to check for the variables that SAS EM would select for the model based on their correlation to the target variable. We ran a regression model after running the variable selection node on default settings.

5.3 Models for target variable will_default

We used decision tree and logistic regression to predict our target variable. Our models utilized different inputs:

- With/without reducing skewness
- With/without imputing for missing values
- Categorizing continuous and nominal variables

5.4 Model Comparison

We used the model comparison node to assess our models. Below is the output of the model comparison node:

Selected Model	Model Node	Model Description	Target Variable	Selection Criterion: Valid: Misclassification Rate	Train: Average Squared Error	Train: Average Error Function	Valid: Mean Square Error
Y	Reg3	Regression (3)	will_default	0.182134	0.139975	0.445361	0.140177
	Reg5	Regression (5)	will_default	0.182173	0.139919	0.445124	0.140059
	Reg2	Regression (2)	will_default	0.182232	0.140199	0.446147	0.140394
	Tree2	Decision Tree (2)	will_default	0.182812	0.149384	.	.
	Tree4	Decision Tree (4)	will_default	0.182812	0.149384	.	.
	Reg	Regression	will_default	0.183068	0.139568	0.442921	0.139673

Fig: 5.8 – Fit statistics for the models for target variable will_default

Below are the ROC curves for the 6 models that we tried.

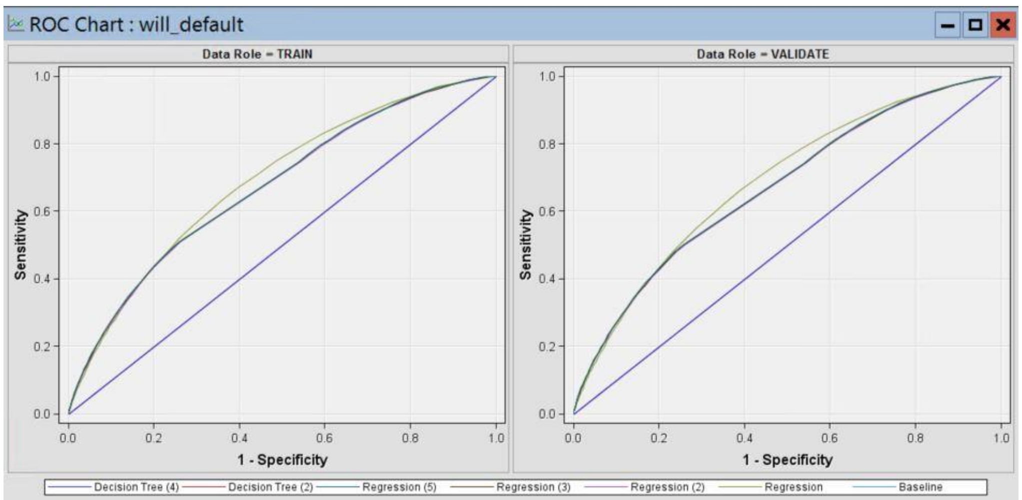


Fig: 5.9 – ROC curves for the models for target variable will_default

The cumulative lift chart for the 6 models is below:

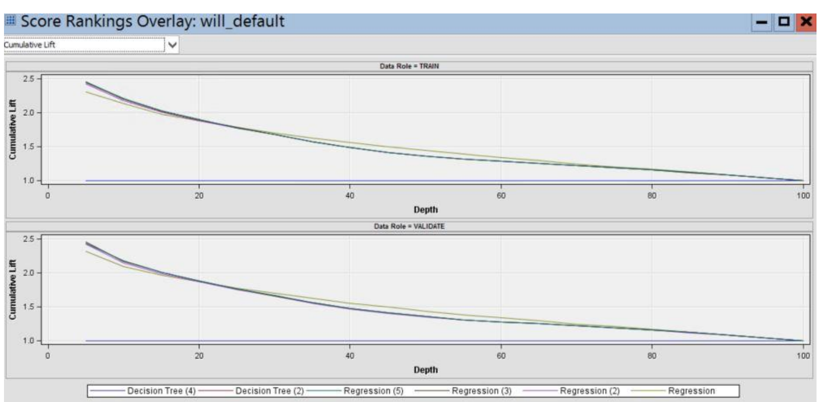


Fig: 5.10 – Cumulative lift chart for the models for target variable will_default

Best Model – Regression (3):

SAS EM chose Regression (3) as the best model for this data. This regression is the one that was used after categorizing some continuous and nominal variables, reducing skewness using Log 10 transformation, and imputing missing values as described in the preprocessing section.

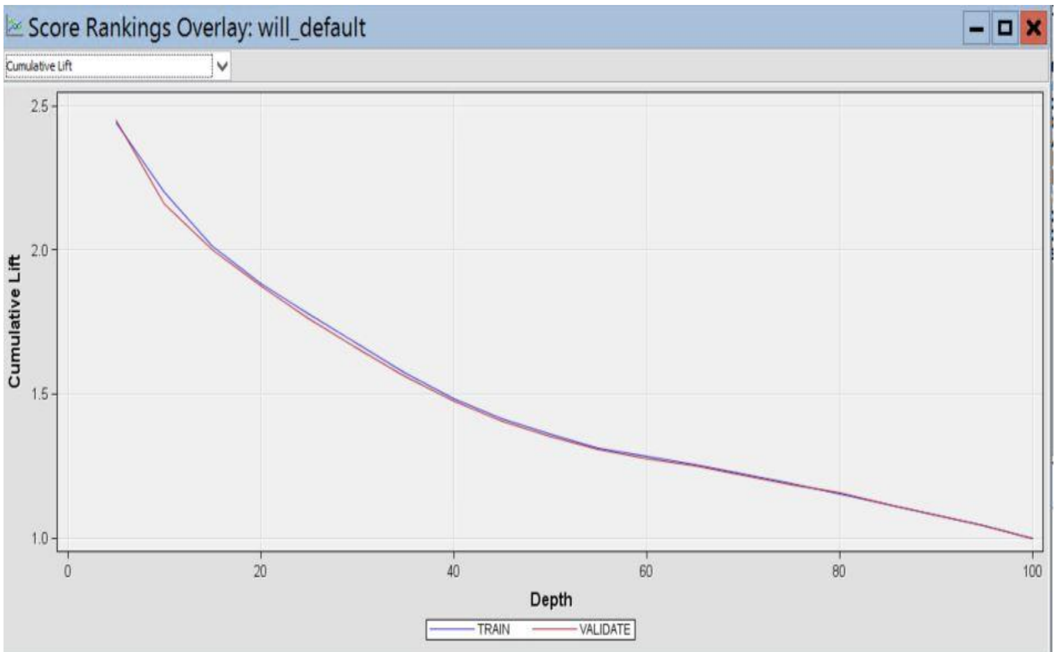


Fig: 5.11 – Cumulative lift chart for the regression (3) model for target variable will_default

Target	Fit Statistics	Statistics Label	Train	Validation
will default	AIC	Akaike's Information Criterion	135952.9	
will default	ASE	Average Squared Error	0.139975	0.14017
will default	AVERR	Average Error Function	0.445361	0.44592
will default	DFE	Degrees of Freedom for Error	152457	
will default	DFM	Model Degrees of Freedom	54	
will default	DFT	Total Degrees of Freedom	152511	
will default	DIV	Divisor for ASE	305022	20335
will default	ERR	Error Function	135844.9	90680.6
will default	FPE	Final Prediction Error	0.140074	
will default	MAX	Maximum Absolute Error	0.988366	0.993
will default	MSE	Mean Square Error	0.140024	0.14017
will default	NOBS	Sum of Frequencies	152511	10167
will default	NW	Number of Estimate Weights	54	
will default	RASE	Root Average Sum of Squares	0.374132	0.37440
will default	RFPE	Root Final Prediction Error	0.374264	
will default	RMSE	Root Mean Squared Error	0.374198	0.37440
will default	SBC	Schwarz's Bayesian Criterion	136489.4	
will default	SSE	Sum of Squared Errors	42695.31	28505.7
will default	SUMW	Sum of Case Weights Times Freq	305022	20335
will default	MISC	Misclassification Rate	0.182216	0.18213

Fig: 5.12 – Fit Statistics for the regression (3) model for target variable will_default

Confusion Matrix:

		Actual - Will_Default	
Predicted - Will_Default		**0 – [No]**	**1 – [Yes]**
	0 – [No]	82819 (True -)	18248 (False -)
	1 – [Yes]	271 (False +)	340 (True +)

Fig: 5.13 – Confusion matrix for the regression (3) model for target variable will_default

5.5 Business implication

When predicting whether the loan will be defaulted on or not, the more expensive error is the false negative. In this case, the model predicts a 0 i.e. loan will be fully paid when actually it will be defaulted on. This will lead to the investor losing both the principal and the profit (interest).

In the false positive error, the model predicts 1 i.e. loan will be defaulted on when it will be fully paid. In this case, the investors lose out on the profit that they would have made by funding the loan.

5.6 Conclusion

This model was designed to help investors decide which loans to invest in. Our best model returned a misclassification rate of 18.22%. This business model helps in the decision making of whether to invest in a loan or not.

6. BI Model – Will_Default – Tracker

We tried to create a model that can be used as a tracker by investors to predict if their investment – a funded loan – will be fully repaid or defaulted on. For this reason, we have included previously rejected variables such as last_pymnt (last payment), tot_pymnt (total payment), tot_received_prncp (total received principal) and tot_received_int (total received interest). This model can be used by investors to track progress and outcome of their loan and plan their own finances accordingly. For instance, if they are relying on the profit from funding the loan and they can predict that the loan will be defaulted on, they can make alternative arrangements for the funds.

6.1 Model Comparison – Will_Default Tracker:

We used the model comparison node to assess our models. Below is the output of the model comparison node:

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Selection Criterion: Valid: Misclassification Rate	Valid: Average Squared Error	Valid: Mean Square Error ▲	Valid: Root Mean Square Error	Train: Misclassification Rate	Train: Average Squared Error	Train: Root Mean Squared Error
Y	Tree2	Tree2	Decision Tree (2)	will_default	0.031492	0.0251...	.	.	0.0303...	0.0241...	.
	Tree4	Tree4	Decision Tree (4)	will_default	0.053276	0.0425...	.	.	0.0515...	0.0415...	.
	Reg3	Reg3	Regression (3)	will_default	0.063288	0.0515...	0.0515...	0.2270...	0.0611...	0.0502...	0.2242...
	Reg2	Reg2	Regression (2)	will_default	0.063347	0.0516...	0.0516...	0.2272...	0.06134	0.0503...	0.2243...
	Reg	Reg	Regression	will_default	0.089803	0.0620...	0.0620...	0.24906	0.0880...	0.0609...	0.2468...
	Reg5	Reg5	Regression (5)	will_default	0.106355	0.0721...	0.0721...	0.2686...	0.1034...	0.0711...	0.2667...

Fig: 6.1 – Fit statistics for the models for target variable will_default tracker

SAS EM chose the decision tree (2) as the best model for this dataset based on the Misclassification rate of the Validation dataset. This decision tree was the model that used the inputs after categorizing the continuous and nominal variables as mentioned in the preprocessing step.

Below are the ROC curves for the 6 models that we tried. The ROC curves show that the decision tree (2) closely follows the left and top border and is the best model for this dataset.

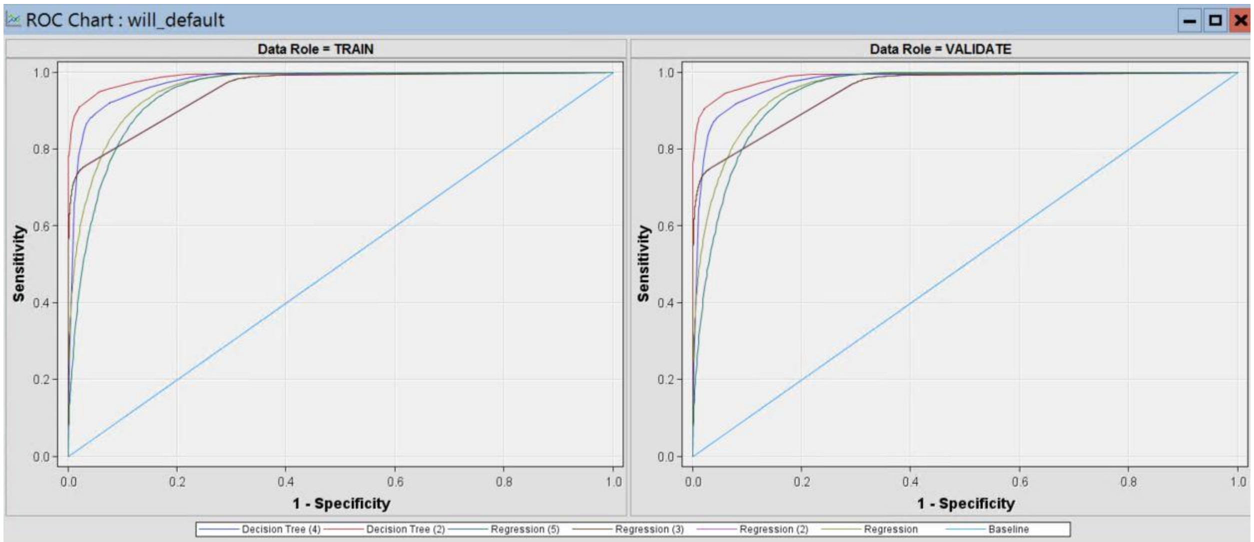


Fig: 6.2 – ROC curves for the models for target variable will_default tracker

The cumulative lift chart for the 6 models that we tried is below:

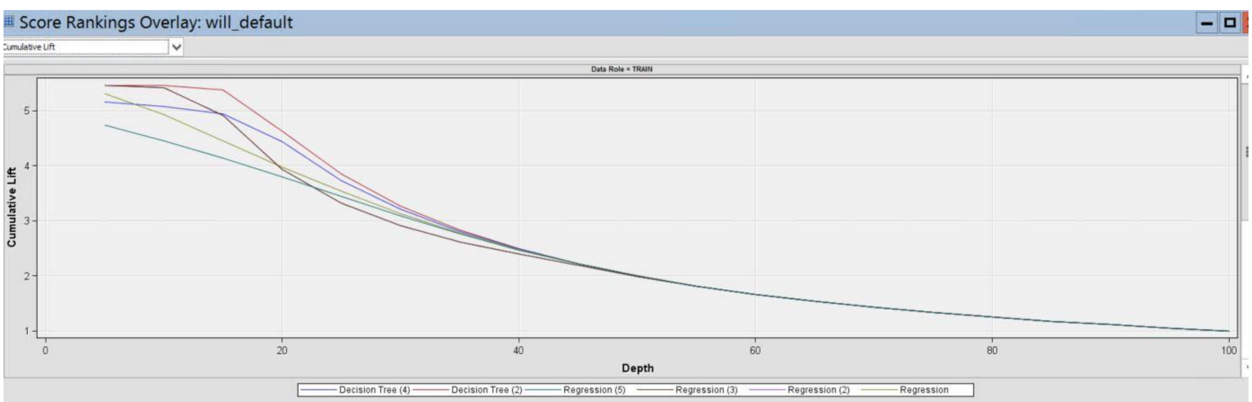


Fig: 6.3 – Cumulative lift chart for the models for target variable will_default tracker

Best Model – Decision Tree (2)

Cumulative Lift Chart and Fit Statistics:

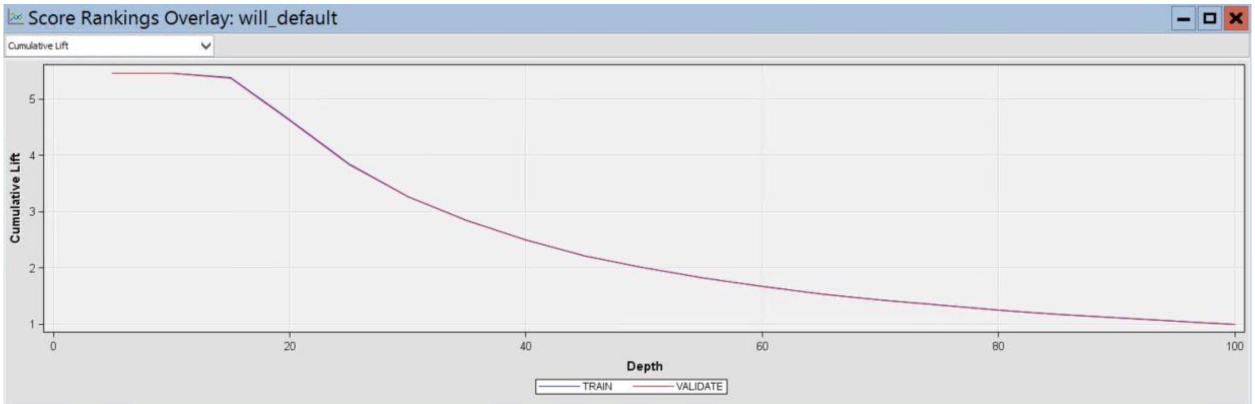


Fig: 6.4 – Cumulative lift chart for the decision tree (2) for target variable will_default tracker

Target	Fit Statistics	Statistics Label	Train	Validation
will_default	NOBS	Sum of Frequencies	152511	101678
will_default	MISC	Misclassification Rate	0.030358	0.031492
will_default	MAX	Maximum Absolute Error	0.999363	0.999363
will_default	SSE	Sum of Squared Errors	7355.488	5111.47
will_default	ASE	Average Squared Error	0.024115	0.025136
will_default	RASE	Root Average Squared Error	0.155289	0.158542
will_default	DIV	Divisor for ASE	305022	203356
will_default	DFT	Total Degrees of Freedom	152511	.

Fig: 6.5 – Fit Statistics for the decision tree (2) for target variable will_default tracker

Confusion Matrix:

		Actual	
		0 – [No]	**1 – [Yes]**
Predicted	**0 – [No]**	82137 (True -)	2249 (False -)
	1 – [Yes]	953 (False +)	16339 (True +)

Fig: 6.6 – Confusion matrix for the decision tree (2) for target variable will_default tracker

6.2 Business implication

When using the tracker model, the more expensive error is the false positive. In the false positive error, the model predicts 1 i.e. loan will be defaulted on when it will be fully paid. In this case, the investors lose out on the opportunity to further invest their profits.

6.3 Conclusion

This model was designed to help investors track the progress and outcome of the loans to invest in. Our best model returned a misclassification rate of 3.03%.

7. References

- www.kaggle.com
- www.lendingclub.com