Analysis of Bank Loan Data

Katie Adamson, Brian Bruno, Colin McCunney, Brandon Rank

MI-353

Dr. Cevikparmak

May 5, 2022

Executive summary

The purpose of this data mining project is to conduct an examination of the dataset obtained from Kaggle concerning bank loans to determine if any significant insights can be gained about the population. Through the use of various algorithms and practices, we hope to uncover useful patterns and information that can be used to assess the state of the bank loan data, as well as enable us to create models which can accurately predict or examine future data. In our examination of the data, which we will discuss later, we first had to explore and prepare our data, followed by the implementation of various models, and finally, a comparison of those created models. Through these important steps, we were able to learn more about the data and the relevant implications of our findings. Overall, we found that all the models performed slightly better than the baseline statistic discussed later in the paper, with a 0.2% improvement upon the Misclassification Rate. In the pages to follow, we will discuss our findings and the relevant information that was examined throughout our analysis.

Project Motivation

The examination of bank loan data presented in this report is important because it provides useful insights into the real world analysis conducted by firms on a daily basis. Banks and other financial institutions, which provide loans to businesses and consumers, are often faced with the question of whether or not a potential applicant will default on their loan. This is an important determination for an institution to make because it will ultimately affect the sustainability and long term success of the firm. Prior to the introduction of data analytic solutions, firms made loan application decisions based upon their decades of experience and risk analysis. With the recent growth and speed of analytics solutions, firms have revolutionized the way in which they determine risk, and subsequently, the way in which they give loans. It is our

goal with this project to uncover how different models can perform with the data we obtained, and to explore the possible methods businesses use to improve their operations.

Data description

As mentioned previously, the data on bank loans utilized in our project was obtained from Kaggle. In total, there were 99,990 records, with over 19,000 records containing missing values as is seen in the class variable summary below. This will potentially negatively affect our analysis due to the fact that missing values can greatly impact the performance of our analytics models, and as such, we will need to carefully consider how we approach the methods used in this project, something we will discuss in the next section.

The data contained a variety of variables, with information regarding the loan status, months since last delinquent payment, loan amount, and much more. These variables contained a combination of both interval and nominal values, as well as two variables that are best described as identifiers. The following is a brief description of each variable and its meaning.

- Load ID - The identification number for each loan
- Customer ID - The identification number for each customer
- Loan Status - Details whether the loan is Fully Paid or Charged Off
- Current Loan Amount - Details the loan amount currently outstanding
- Term - Details whether the loan is a short term or long term loan
- Credit Score - Provides the customers credit score
- Annual Income - Provides the customers annual income
- Years in Current Job - Details customers years of employment in current position
- Home Ownership - Details the level of ownership the customer has for their home
- Purpose - Describes the purpose of the loan

- Monthly Debt - Describes the monthly debt for the loan and customer
- Years of Credit History - Provides the number of years a customer has of credit history
- Months Since Last Delinquent - Provides the number of months since a late loan payment
- Number of Open Accounts - Provides the number of accounts a customer has open
- Number of Credit Problems - Provides the number of credit issues a customer has
- Current Credit Balance - Provides the current balance owed by a customer
- Maximum Open Credit - Provides the maximum amount a customer can borrow
- Bankruptcies - Details how many times a customer has experienced bankruptcy
- Tax Liens - Details how many tax liens a customer has

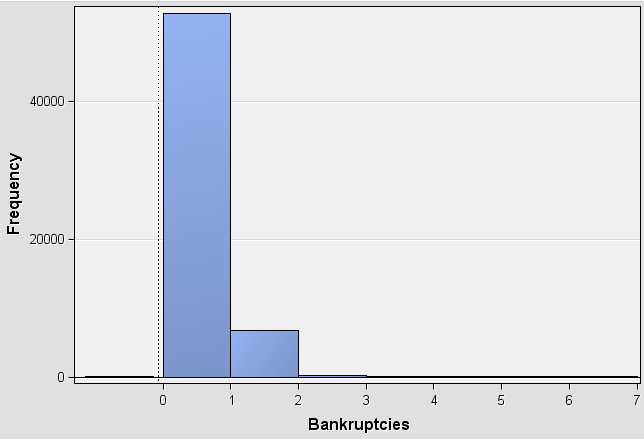
The following is an image of the "Edit Variables" dialogue box, and displays the chosen data type for each variable in the dataset that we just described.

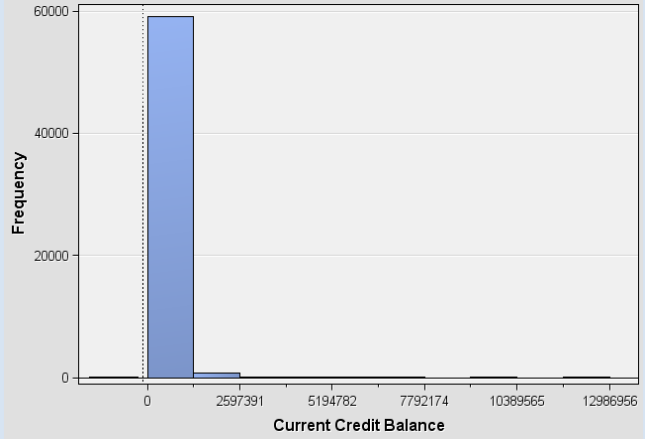
Name	Role	Level	Report	Order	Drop	Lower Limit	Upper Limit
Annual_Income	Input	Interval	No		No	.	.
Bankruptcies	Input	Interval	No		No	.	.
Credit_Score	Input	Interval	No		No	.	.
Current_Credit_	Input	Interval	No		No	.	.
Current_Loan_A	Input	Interval	No		No	.	.
Customer_ID	ID	Nominal	No		No	.	.
Home_Ownershi	Input	Nominal	No		No	.	.
Loan_ID	ID	Nominal	No		No	.	.
Loan_Status	Target	Nominal	No		No	.	.
Maximum_Open_	Input	Interval	No		No	.	.
Monthly_Debt	Input	Interval	No		No	.	.
Months_since_la	Input	Interval	No		No	.	.
Number_of_Cre	Input	Interval	No		No	.	.
Number_of_Ope	Input	Interval	No		No	.	.
Purpose	Input	Nominal	No		No	.	.
Tax_Liens	Input	Interval	No		No	.	.
Term	Input	Nominal	No		No	.	.
Years_in_curren	Input	Nominal	No		No	.	.
Years_of_Credit	Input	Interval	No		No	.	.

After looking at each variable on the surface level and gaining an understanding of its title, we then conducted a variable exploration of our variables. We first looked at each interval variable under the "explore" option in the file import node's "edit variable" option box. The following is our results from that exploration. We want to draw attention to the high degree of

skewness present within the data. We believe that this is the result of the nature of the data, as bank loans are given not just to individual consumers, but also to small and large businesses. Because of this, the loans given within this dataset may contain individuals and organizations with vastly different credit histories and levels, allowing for the data to contain extremely high values that skew the data positively. While these high values may be of importance, due to the fact that they represent true loan data from our data source, we elected to filter out these values so as to obtain a more normal distribution for the data.

The bankruptcies variable appears to be positively skewed, having a maximum value of 7.0, and a minimum value of 0.0. The majority of loans, 52,792 to be exact, had 0 bankruptcies, distantly followed by 6,718 loans with 1 bankruptcy. The remaining loans had between 2-7 bankruptcies.

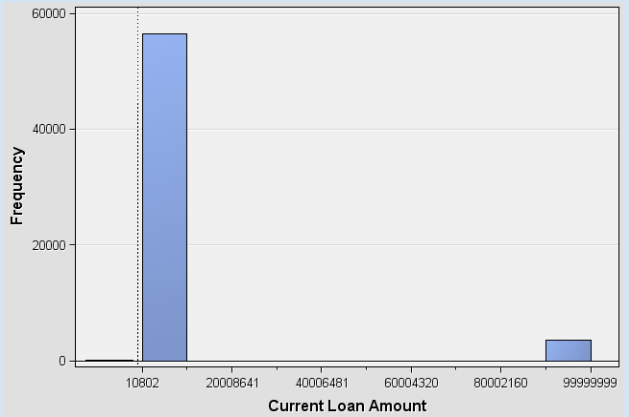


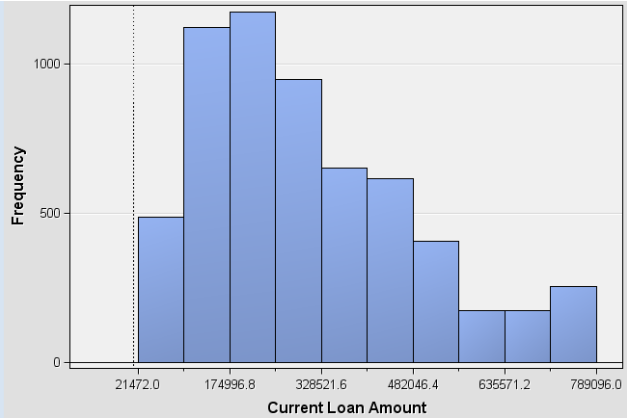


Current Credit Balance appears to be positively skewed that has several large data points that cause it to be positively skewed. Furthermore, Current Credit Balance had a maximum value of 32,878,968.00, and a minimum value of 0.0.

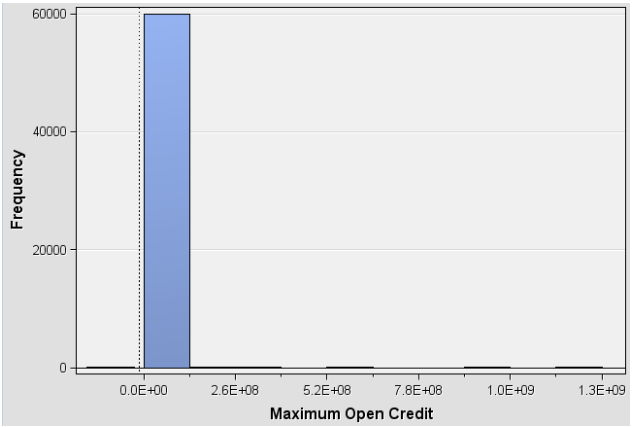
Current Loan Amount appears to be positively skewed, and it has a maximum value of 99,999,999.00, and a minimum value of 10802. When only looking at the default setting, which

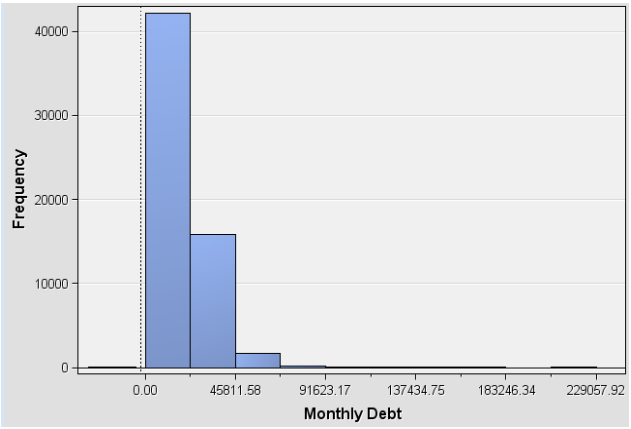
is simply the “top” 6,000 records, we see what appears to be a somewhat well distributed variable, still skewed right, but nevertheless, not extreme. However, when we look at the variable as a whole, the 3,560 records with a current loan amount of 99,999,999 drastically modifies the distribution. This is something we will keep in mind going forward, and may ultimately filter depending on the results we gather. The two distributions can be seen together below, with the distribution on the left containing 60,000 rows, while the distribution on the right contains only 6,000.





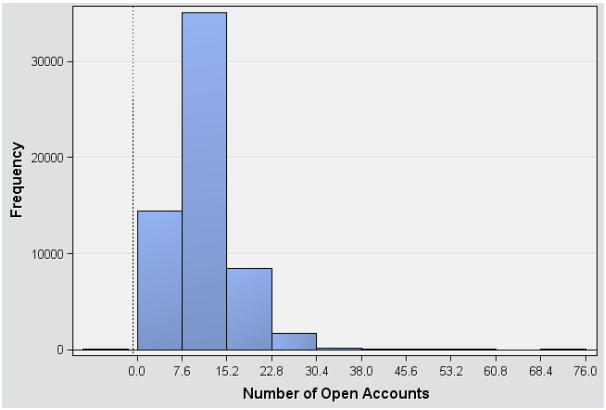
Maximum Open Credit appears to have an irregular distribution and it appears to have multiple outliers that will need to be filtered out. Maximum Open Credit had a maximum value of 1,539,737,892.00, and a minimum value of 0.0. Again, we believe that the reason for such a high maximum is because businesses have the ability to borrow much larger sums of money, resulting in such a highly skewed distribution.





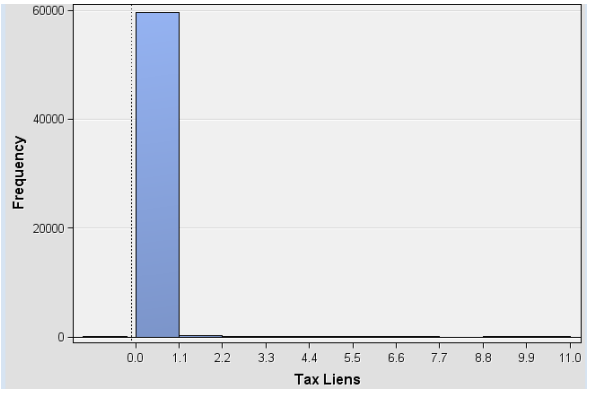
Monthly Debt appears to be positively skewed with a few outliers at the end of the tail and has a maximum value of 435843.28, and a minimum value of 0.0. The histogram does not reach this maximum due to the number of records exceeding the 60,000 fetched.

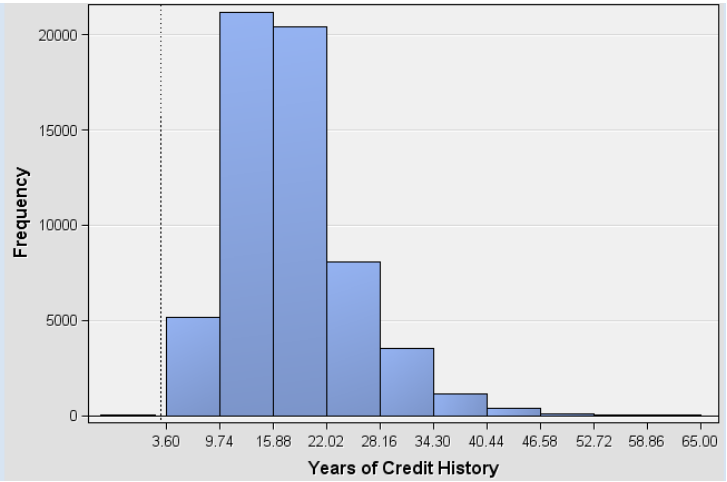
Number of Credit Problems appears to be positively skewed with a few outliers at the end of the tails and it has a maximum value of 15.00, and a minimum value of 0.0. The explore feature was not able to display all of the values, resulting in the histogram only reaching a maximum of 12.



Number of Open Accounts appears to be slightly positively skewed with some larger values trailing off at the end of its tail. Number of Open Accounts had a maximum value of 76.00, and a minimum value of 0.0.

Tax Liens appear to have an irregular distribution with several outliers. Tax Liens had a maximum value of 15.00, and a minimum value of 0.0, but the extreme majority of loans were customers with 0 tax liens.





Years of Credit History appears to be slightly positively skewed with roughly one outlier. Years of Credit History has a maximum value of 70.50, and a minimum value of 3.6. After running the summary statistics, there were multiple missing values for the class and interval variables.

After doing this initial exploration from the "Edit Variables" dialogue box, we then completed a further analysis by using SAS's StatExplore node. From that node, we were able to look at the interval and nominal variables, gaining an understanding of the proportion of loans of each status, the number of missing values present in the data, as well as the skewness levels and summary statistics like the mean, minimum and maximum values. The first image presented here displays the output data concerning the distribution of our target variable, "Loan Status".

Distribution of Class Target and Segment Variables
(maximum 500 observations printed)

Data Role=TRAIN

Data Role	Variable Name	Role	Level	Frequency Count	Percent
TRAIN	Loan_Status	TARGET	Fully Paid	77353	77.3607
TRAIN	Loan_Status	TARGET	Charged Off	22636	22.6383
TRAIN	Loan_Status	TARGET		1	0.0010

From the image above, we can see that the distribution of the data tends towards the "Fully Paid" loan status, with 77.36% of the data containing this label, while only 22.63% of the data is of the "Charged Off" label. This will be important to consider during our analysis because there is a much greater proportion of loans of the "Fully Paid" status than are of the "Charged

Off" status. In addition, this provides us with our baseline model for which we will compare our chosen models later. The error rate that we are trying to improve upon is the proportion of the data that is of the class label "Charged Off", or 22.6383%. The following image details the overall interval variable statistics from the output.

Interval Variable Summary Statistics
(maximum 500 observations printed)

Data Role=TRAIN

Variable	Role	Mean	Standard Deviation	Non Missing	Missing	Minimum	Median	Maximum	Skewness	Kurtosis
Annual_Income	INPUT	1378300	1081380	80837	19153	76627	1174181	1.6556E8	46.89133	6624.427
Bankruptcies	INPUT	0.117714	0.351399	99784	206	0	0	7	3.506559	18.5358
Credit_Score	INPUT	716.2927	28.29736	80837	19153	585	722	751	-1.33806	2.016052
Current_Credit_Balance	INPUT	294638.7	376187.9	99988	2	0	209798	32878968	14.1542	697.4555
Current_Loan_Amount	INPUT	11760718	31784257	99989	1	10802	312246	99999999	2.415946	3.83717
Maximum_Open_Credit	INPUT	760814.5	8385006	99986	4	0	467874	1.5397E9	132.631	20392.4
Monthly_Debt	INPUT	18473.03	12175.11	99988	2	0	16221.44	435843.3	2.214031	22.19446
Months_since_last_delinquent	INPUT	34.90033	21.99731	46853	53137	0	32	176	0.434379	-0.74571
Number_of_Credit_Problems	INPUT	0.16827	0.482668	99988	2	0	0	15	4.824308	48.03319
Number_of_Open_Accounts	INPUT	11.12836	5.009811	99988	2	0	10	76	1.179354	3.043632
Tax_Liens	INPUT	0.029316	0.258198	99978	12	0	0	15	15.49928	402.0185
Years_of_Credit_History	INPUT	18.19905	7.015517	99988	2	3.6	16.9	70.5	1.071612	1.74071

As we presented graphically in our initial exploration of these variables, the interval data contained a large number of missing values for some variables, as well as a high degree of skewness for some variables. For example, the variable "Months Since Last Delinquent", contained 51,137 missing values, followed by "Annual Income" and "Credit Score", each with 19,153 missing values. This highlights the issue we face with regards to missing values, and explains the need for further action that we will discuss in the data preparation portion of this report. In addition to the missing values, we can see from this output that many variables, like "Annual Income" and "Current Credit Balance" have high skewness values, 46.89 and 14.15 respectively. With many other variables also having a skewness greater than 1, thus indicating that the data is not normally distributed. Again, we believe this to be the result of the loan data possibly containing both individuals and businesses, resulting in this discrepancy in values. For these three variables with a high number of missing values, we also wanted to explore where the

majority of these missing values fit in the data, and more specifically, whether these values were generally for loans with the status “Fully Paid” or for loans of the “Charged Off” status.

Data Role=TRAIN Variable=Annual_Income

Target	Target Level	Median	Missing	Non Missing	Minimum	Maximum	Mean	Standard Deviation	Skewness	Kurtosis	Role	Label
Loan_Status		.	1	0	.	.	.	.	.	.	INPUT	Annual Income
Loan_Status	Charged Off	1085812	5428	17208	76627	1.6556E8	1267872	1506319	76.73152	8240.542	INPUT	Annual Income
Loan_Status	Fully Paid	1213017	13724	63629	81092	36475440	1408164	931579.5	5.619375	90.34965	INPUT	Annual Income
OVERALL		1174181	19153	80837	76627	1.6556E8	1378300	1081380	46.89133	6624.427	INPUT	Annual Income

Data Role=TRAIN Variable=Credit_Score

Target	Target Level	Median	Missing	Non Missing	Minimum	Maximum	Mean	Standard Deviation	Skewness	Kurtosis	Role	Label
Loan_Status		.	1	0	.	.	.	.	.	.	INPUT	Credit Score
Loan_Status	Charged Off	719	5428	17208	585	751	710.3911	31.26032	-1.20737	1.211814	INPUT	Credit Score
Loan_Status	Fully Paid	723	13724	63629	585	751	717.8887	27.2225	-1.35639	2.227754	INPUT	Credit Score
OVERALL		722	19153	80837	585	751	716.2927	28.29736	-1.33806	2.016052	INPUT	Credit Score

Data Role=TRAIN Variable=Months_since_last_delinquent

Target	Target Level	Median	Missing	Non Missing	Minimum	Maximum	Mean	Standard Deviation	Skewness	Kurtosis	Role	Label
Loan_Status		.	1	0	.	.	.	.	.	.	INPUT	Months since last delinquent
Loan_Status	Charged Off	31	12227	10409	0	152	34.33923	22.24143	0.447719	-0.76489	INPUT	Months since last delinquent
Loan_Status	Fully Paid	32	40909	36444	0	176	35.06059	21.92475	0.431189	-0.73959	INPUT	Months since last delinquent
OVERALL		32	53137	46853	0	176	34.90033	21.99731	0.434379	-0.74571	INPUT	Months since last delinquent

From the images above, we can see that primarily, the missing values were from loans with the “Fully Paid” Status. Interestingly, approximately 75% of the missing values for each of these three variables is attributable to the “Fully Paid” loans, a proportion that is very similar to our distribution of the “Loan Status Variable”. For the variables “Annual Income” and “Credit Score”, there are 19,153 records with missing values for these variables, 13,724 of which are of the loan status “Fully Paid”, while 5,428 are of the status “Charged Off”. It is interesting that these variables have an identical number of missing values in both class cases, and as such, we believe that these values represent identical records for which there was not sufficient data to complete the record. The “Months Since Last Delinquent” variable contained 53,137 missing values, 40,909 of which were for “Fully Paid” loans, and 12,227 of which were for “Charged Off” loans.

After looking at our interval variables, we then shifted our focus to the nominal variables in our data. From the following four tables, we can see that the variable “Years in Current Job” contained nearly all of the missing values for the nominal variables, with 4,222 records having no value present. Another interesting item that we noticed was that for these variables, the Mode, or class most commonly present in the variables was the same for loans that were “Fully Paid” and “Charged Off”, with the exception of “Home Ownership”, where loans of the “Fully Paid” status were primarily home mortgages, whereas loans of the “Charged Off” Status were most commonly for renting.

Class Variable Summary Statistics by Class Target
(maximum 500 observations printed)

Data Role=TRAIN Variable Name=Home_Ownership

Target	Target Level	Number of Levels	Missing	Mode	Mode Percentage	Mode2	Mode2 Percentage
Loan_Status		1	1		100.0		0.00
Loan_Status	Charged Off	4	0	Rent	46.81	Home Mortgage	43.78
Loan_Status	Fully Paid	5	0	Home Mortgage	49.77	Rent	40.84
OVERALL		6	1	Home Mortgage	48.41	Rent	42.19

Data Role=TRAIN Variable Name=Purpose

Target	Target Level	Number of Levels	Missing	Mode	Mode Percentage	Mode2	Mode2 Percentage
Loan_Status		1	1		100.0		0.00
Loan_Status	Charged Off	16	0	Debt Consolidation	79.15	other	6.30
Loan_Status	Fully Paid	17	1	Debt Consolidation	78.38	Home Improvements	6.05
OVERALL		17	2	Debt Consolidation	78.55	other	6.04

Data Role=TRAIN Variable Name=Term

Target	Target Level	Number of Levels	Missing	Mode	Mode Percentage	Mode2	Mode2 Percentage
Loan_Status		1	1		100.0		0.00
Loan_Status	Charged Off	2	0	Short Term	63.04	Long Term	36.96
Loan_Status	Fully Paid	2	0	Short Term	74.89	Long Term	25.11
OVERALL		3	1	Short Term	72.21	Long Term	27.79

Data Role=TRAIN Variable Name=Years_in_current_job

Target	Target Level	Number of Levels	Missing	Mode	Mode Percentage	Mode2	Mode2 Percentage
Loan_Status		1	1		100.0		0.00
Loan_Status	Charged Off	12	1271	10+ years	29.70	2 years	8.84
Loan_Status	Fully Paid	12	2950	10+ years	31.54	2 years	9.22
OVERALL		12	4222	10+ years	31.12	2 years	9.13

Data Preparation

In preparing the data for use in our project, we first went over our exploration of the data. As we noted in the previous section, there were a large number of missing values for various variables, in addition to heavily skewed data, and in particular, one record and one variable that required attention. For starters, the individual record that we focused on contained formatting inconsistent with the rest of the dataset, and as a result, we elected to remove this record in order to prevent any issues in the future.

When cleaning the data, we focused on the credit score variable. The reason for this was because the original data source had two different entry methods for the credit scores, one being the normal, three digit scores, such as 756, and the other containing four digit scores, for example, 7560. Because of this added digit, cleaning this variable was essential to obtaining strong analytical results, as SAS EM does not know to interpret the four digit scores as the same as the three digit scores. In order to modify the credit score column to contain all three digit codes, we first copied the data in excel into a separate worksheet, and then utilized the LEFT function to pull only the first three digits of each credit score. After applying this to all

observations, we replaced the existing credit score data with the new, cleaned data. Seen below is an example of the data before being cleaned, and after.

Credit Score_Raw	Credit Score - Left Function
709	709
741	741
721	721
7290	729
730	730

Our next focus was on the issue of the missing values in the data set. In this stage, we were not so much concerned with removing these values, but rather, with providing a value for these instances that would allow SAS to recognize them as missing values. In the raw excel data, there was a mixture of blank cells, as well as a very large number of cells containing the value "NA" or "n/a". Unfortunately, SAS cannot properly interpret these values, and in many cases, will interpret "NA" or "n/a" as another class label or value for the interval variables. Because we want these cells to indicate missing values, and not additional class labels or values, we needed to address this issue. To remedy this issue, we utilized the built in excel feature that enables us to replace all instances in which these values were present with a blank cell, thereby removing this issue before we imported the data into SAS. After completing this action, we were able to obtain a dataset which contained true "missing" values.

After this, we imported our data into SAS EM and began to address the missing values and outliers we previously discussed directly. We first used the filter node to remove only the outliers, choosing not to also filter the missing values directly. This process removed 10,695 records from our data, leaving us with 89,295 records to use in our analysis. The reason for this was because after our initial experimentation with the data, we found that filtering all of the

missing values was removing a very large portion of the data, approximately 27,000 records. As such, we wanted to utilize SAS EM's impute node so as to retain the number of records in the data, while also filling those missing values with representative values from the dataset.

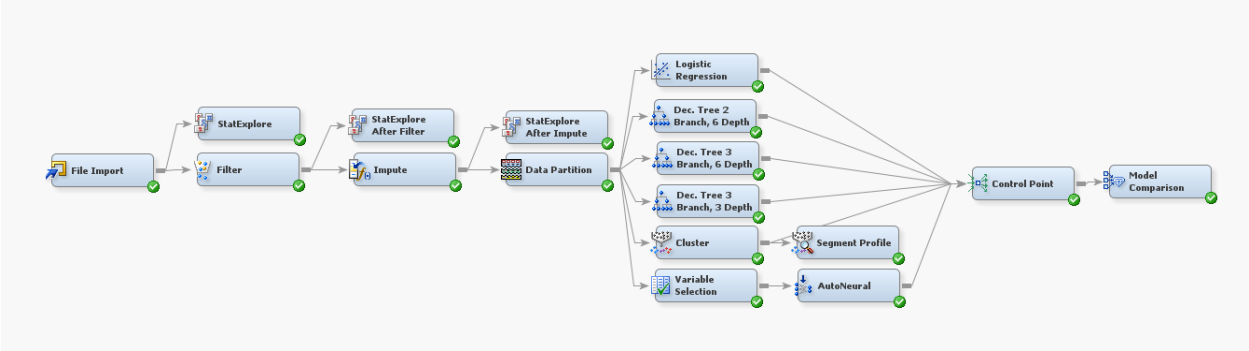
We did not impute the values prior to filtering out the outliers because we did not want the outliers present in the data to pull the imputed values, in this case, the means for our interval variables, up. By filtering the outliers and then imputing the values, we filled in the missing values with the most representative data from the dataset. When setting up our imputation node, we elected to use the "Mean" as the default input method for the missing interval values, and the "Count" as the default input method for the class values. After running the impute node, all of the missing values were replaced by their imputed values.

The final step in our data preparation was to partition the data so as to allow for the models to train and validate themselves. We decided to go with 70% of the data as our training set, and 30% as our validation set. This was because we had such a large difference in the proportion of our loan statuses that we wanted to ensure the training data was sufficient to produce strong models.

Modeling in SAS Enterprise Miner

In choosing what models to run for our analysis, we needed to consider the fact that our chosen target variable, "Loan Status", was a categorical variable with two levels, "Fully Paid" and "Charged Off". This is significant because it prevents us from running certain models, for example, a linear regression, which requires an interval response variable. In our case, this meant that we would need to utilize a logistic regression as opposed to a linear regression. In the end, we decided to run a logistic regression model, three different decision tree variations, a clustering analysis and finally, an auto neural network. After running these models we then connected them

via a control point to the “Model Comparison” node, which allowed us to determine which model was producing the best results of the group, as well as to compare them to the baseline in our analysis, which is the initial distribution of the data.



Logistic Regression

The first model we chose to run for our examination of the data was a logistic regression. To do this, we first selected the regression node, and then changed the regression type to logistic, changing no other settings for the model. After running the logistic regression, the results showed that the Misclassification Rate for the model was 22.4382% for the training set, and 22.4346% for the validation set. In addition, the Average Square Error for the training set was 16.263%, with a validating error of 16.3053%.

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation
Loan_Status	Loan Status	_AIC_	Akaike's Information C...	61088.08	.
Loan_Status	Loan Status	_ASE_	Average Squared Error	0.16263	0.163053
Loan_Status	Loan Status	_AVERR_	Average Error Function	0.488202	0.489918
Loan_Status	Loan Status	_DFE_	Degrees of Freedom f...	62476	.
Loan_Status	Loan Status	_DFM_	Model Degrees of Fre...	29	.
Loan_Status	Loan Status	_DFT_	Total Degrees of Free...	62505	.
Loan_Status	Loan Status	_DIV_	Divisor for ASE	125010	53578
Loan_Status	Loan Status	_ERR_	Error Function	61030.08	26248.83
Loan_Status	Loan Status	_FPE_	Final Prediction Error	0.162781	.
Loan_Status	Loan Status	_MAX_	Maximum Absolute Err...	0.997573	0.998164
Loan_Status	Loan Status	_MSE_	Mean Square Error	0.162705	0.163053
Loan_Status	Loan Status	_NOBS_	Sum of Frequencies	62505	26789
Loan_Status	Loan Status	_NW_	Number of Estimate ...	29	.
Loan_Status	Loan Status	_RASE_	Root Average Sum of ...	0.403274	0.403798
Loan_Status	Loan Status	_RFPE_	Root Final Prediction ...	0.403461	.
Loan_Status	Loan Status	_RMSE_	Root Mean Squared E...	0.403368	0.403798
Loan_Status	Loan Status	_SBC_	Schwarz's Bayesian C...	61350.33	.
Loan_Status	Loan Status	_SSE_	Sum of Squared Errors	20330.37	8736.058
Loan_Status	Loan Status	_SUMW_	Sum of Case Weights ...	125010	53578
Loan_Status	Loan Status	_MISC_	Misclassification Rate	0.224382	0.224346

We also looked at the event classification table for the logistic regression to look at how the model was doing with respect to accurately predicting the class variable. We found, as is displayed in the image below, that while the training and validation set were achieving a high number of true positives, in this case, accurately predicting the class as “Fully Paid”, the model was also sustaining a very high number of false positives.

Event Classification Table

Data Role=TRAIN Target=Loan_Status Target Label=Loan Status

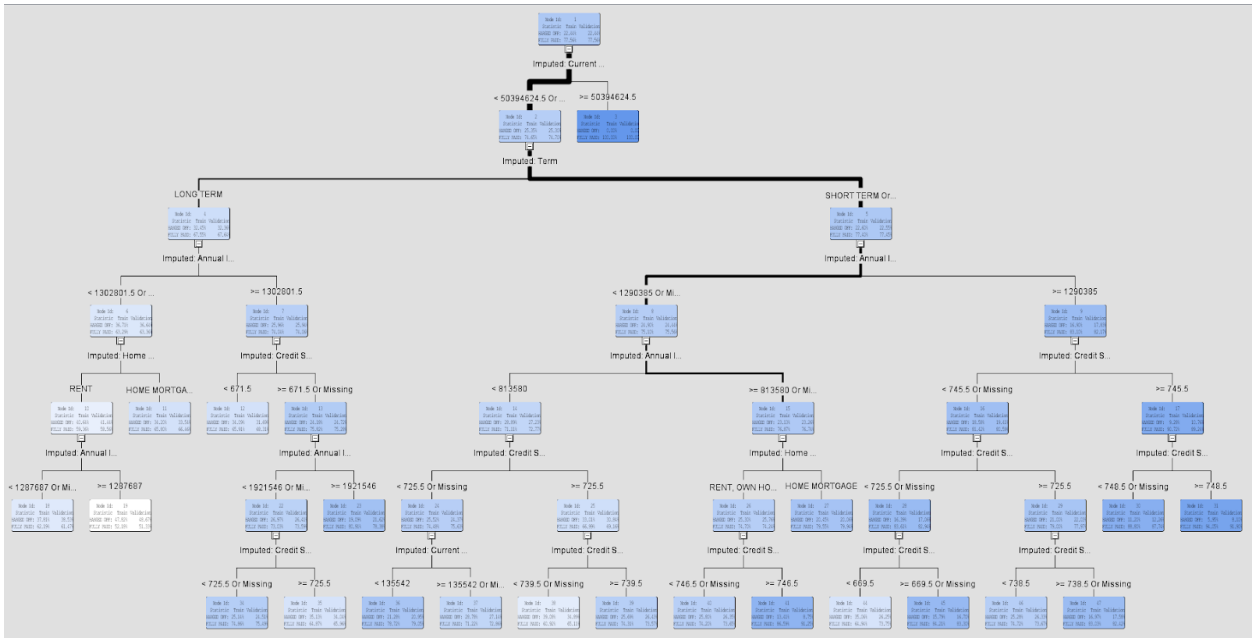
False Negative	True Negative	False Positive	True Positive
11	13	14015	48467

Data Role=VALIDATE Target=Loan_Status Target Label=Loan Status

False Negative	True Negative	False Positive	True Positive
7	9	6003	20770

Decision Tree 1: 2 Branch Maximum, 6 Depth Maximum

The next model we ran in SAS was our first of three decision trees. For this tree, we decided to retain the default SAS settings, with the tree having a maximum branch of 2 and depth of 6, however, we changed the assessment measure to Average Square Error.



This first decision tree had a Misclassification Rate of 22.4414% for the training set and 22.442% for the validation set, as well as an Average Squared Error of 16.1942% and 16.2928% respectively.

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation
Loan_Status	Loan Status	_NOBS_	Sum of Frequencies	62505	26789
Loan_Status	Loan Status	_MISC_	Misclassification Rate	0.224414	0.22442
Loan_Status	Loan Status	_MAX_	Maximum Absolute Err...	0.940476	0.940476
Loan_Status	Loan Status	_SSE_	Sum of Squared Errors	20244.33	8729.38
Loan_Status	Loan Status	_ASE_	Average Squared Error	0.161942	0.162928
Loan_Status	Loan Status	_RASE_	Root Average Squared...	0.40242	0.403644
Loan_Status	Loan Status	_DIV_	Divisor for ASE	125010	53578
Loan_Status	Loan Status	_DFT_	Total Degrees of Free...	62505	.

From the “Tree” and “Output” cards, we determined that the first split occurred based on the variable “Imputed: Current Loan Amount”, which had a variable importance of 1.0000 in both the training and validation sets. The following two variables with the highest importance were the “Imputed Annual Income” and “Imputed: Term”, with Annual Income having a higher training importance, and Term having a higher validation importance. In total, there were 21 leaf nodes produced as a result of this first decision tree model.

Variable Name	Label	Number of Splitting Rules	Importance	Validation Importance	Ratio of Validation to Training Importance
IMP_Current_Loan_Amount	Imputed: Current Loan Amount	2	1.0000	1.0000	1.0000
IMP_Annual_Income	Imputed: Annual Income	5	0.5589	0.4803	0.8593
IMP_Term	Imputed: Term	1	0.5103	0.5149	1.0090
IMP_Credit_Score	Imputed: Credit Score	10	0.4497	0.3841	0.8542
IMP_Home_Ownership	Imputed: Home Ownership	2	0.2233	0.2679	1.1993

Our final findings from the first decision tree showed that once again, the model was doing poorly at accurately predicting loans with the class “Charged Off”, with every instance being wrongly classified as a “Fully Paid” loan.

Event Classification Table

Data Role=TRAIN Target=Loan_Status Target Label=Loan Status

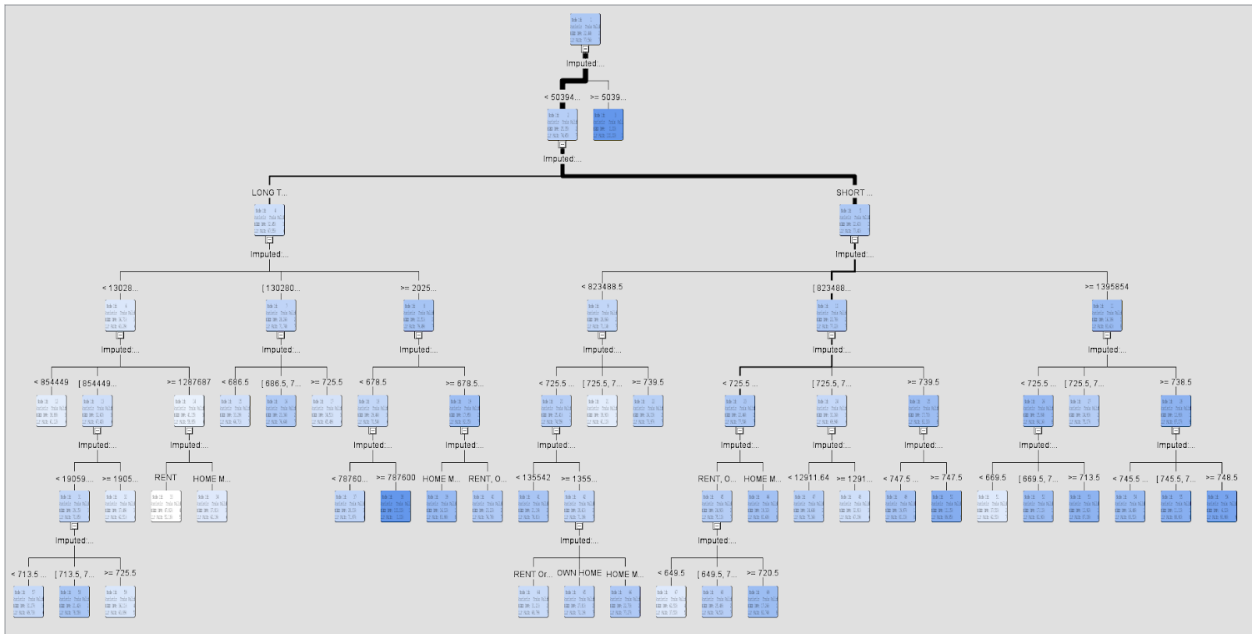
False Negative	True Negative	False Positive	True Positive
0	0	14028	48478

Data Role=VALIDATE Target=Loan_Status Target Label=Loan Status

False Negative	True Negative	False Positive	True Positive
0	0	6012	20777

Decision Tree 2: 3 Branch Maximum, 6 Depth Maximum

The third model we ran in SAS was our second decision tree. For this tree, we decided to retain the settings of the first tree, but this time, we increased the maximum allowed branch to 3. The tree shown below had a total of 36 leaf nodes, which is 15 more than the previous tree which only allowed a maximum branch of 2.



The second decision tree had a Misclassification Rate of 22.4158% and 22.4458% for the training and validation sets, respectively. In addition, the Average Squared Error was 16.0913% and 16.2289% for the training and validation sets.

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation
Loan_Status	Loan Status	_NOBS_	Sum of Frequencies	62505	26789
Loan_Status	Loan Status	_MISC_	Misclassification Rate	0.224158	0.224458
Loan_Status	Loan Status	_MAX_	Maximum Absolute Err...	0.939794	0.939794
Loan_Status	Loan Status	_SSE_	Sum of Squared Errors	20115.76	8695.137
Loan_Status	Loan Status	_ASE_	Average Squared Error	0.160913	0.162289
Loan_Status	Loan Status	_RASE_	Root Average Squared...	0.40114	0.402851
Loan_Status	Loan Status	_DIV_	Divisor for ASE	125010	53578
Loan_Status	Loan Status	_DFT_	Total Degrees of Free...	62505	.

We determined that the first split in the decision tree was once again with the variable “Imputed: Current Loan Amount”, followed in the second split by the “Imputed: Term”. The Current Loan Amount variable once again had an importance of 1.0000 for both the training and validation sets, but we see that with the remaining variables listed, the importances slightly changed.

Variable Name	Label	Number of Splitting Rules	Importance	Validation Importance	Ratio of Validation to Training Importance
IMP_Current_Loan_Amount	Imputed: Current Loan Amount	3	1.0000	1.0000	1.0000
IMP_Annual_Income	Imputed: Annual Income	3	0.5711	0.4737	0.8295
IMP_Credit_Score	Imputed: Credit Score	10	0.5405	0.4867	0.9003
IMP_Term	Imputed: Term	1	0.5078	0.5142	1.0126
IMP_Home_Ownership	Imputed: Home Ownership	4	0.2362	0.2698	1.1419
IMP_Monthly_Debt	Imputed: Monthly Debt	2	0.1693	0.1021	0.6029

After looking at the classification table below, we continued to see that the model produced a high number of false positives, but when compared with the first decision tree, it produced a few true negatives, 36 instances, and a few false negatives, 21 instances.

Event Classification Table

Data Role=TRAIN Target=Loan_Status Target Label=Loan Status

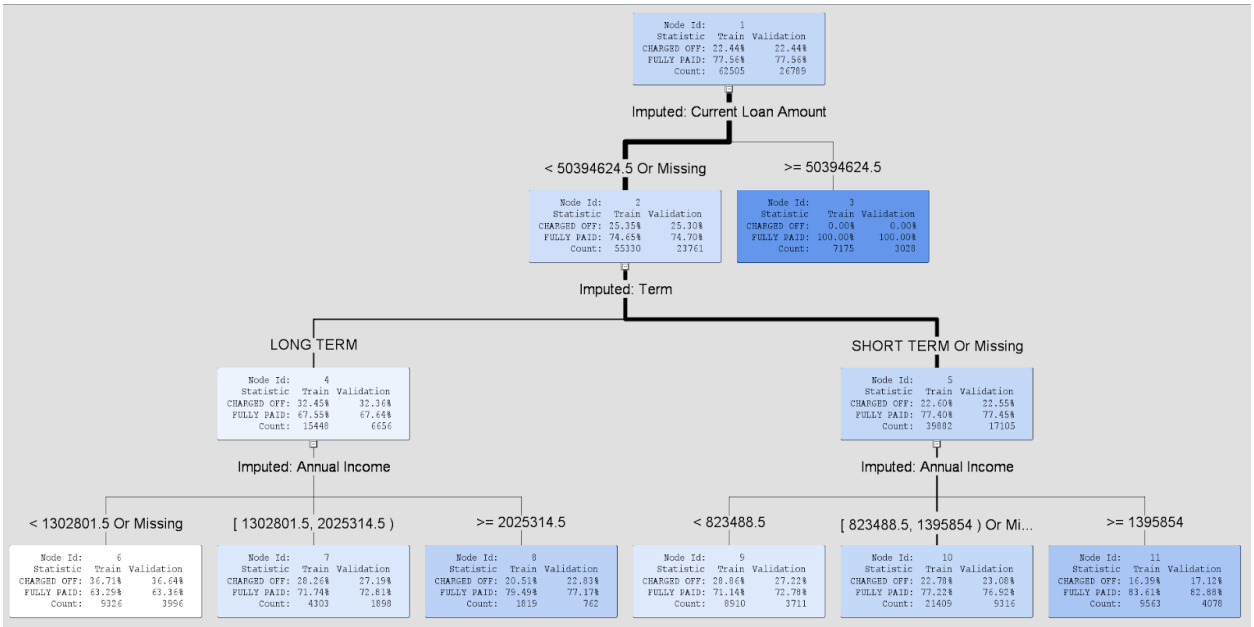
False Negative	True Negative	False Positive	True Positive
12	28	14000	48466

Data Role=VALIDATE Target=Loan_Status Target Label=Loan Status

False Negative	True Negative	False Positive	True Positive
9	8	6004	20768

Decision Tree 3: 3 Branch Maximum, 3 Depth Maximum

Out of curiosity, we decided to run a final decision tree model, with this final model continuing on the path we took in the second tree, but this time, restricting the maximum depth of the tree to 3. We did this with the goal of attempting to produce a tree which might utilize different splitting criteria to arrive at the optimal tree.



This third tree had a Misclassification Rate of 22.4414% and 22.442% for the training and validation sets, as well as an Average Squared Error of 16.3837% and 16.4508%, respectively.

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation
Loan_Status	Loan Status	_NOBS_	Sum of Frequencies	62505	26789
Loan_Status	Loan Status	_MISC_	Misclassification Rate	0.224414	0.22442
Loan_Status	Loan Status	_MAX_	Maximum Absolute Err...	0.836139	0.836139
Loan_Status	Loan Status	_SSE_	Sum of Squared Errors	20481.21	8814.025
Loan_Status	Loan Status	_ASE_	Average Squared Error	0.163837	0.164508
Loan_Status	Loan Status	_RASE_	Root Average Squared...	0.404767	0.405596
Loan_Status	Loan Status	_DIV_	Divisor for ASE	125010	53578
Loan_Status	Loan Status	_DFT_	Total Degrees of Free...	62505	.

The order of the splits was shown in the decision tree above, with the first split being based on the variable “Imputed: Current Loan Amount”, as it was in the first two trees, followed by the term and annual income. The variable importances were once again similar to the original, however, we again saw the training and validation importances flip for the annual income and term.

Variable Importance

Variable Name	Label	Number of Splitting Rules	Importance	Validation Importance	Ratio of Validation to Training Importance
IMP_Current_Loan_Amount	Imputed: Current Loan Amount	1	1.0000	1.0000	1.0000
IMP_Annual_Income	Imputed: Annual Income	2	0.5474	0.4686	0.8561
IMP_Term	Imputed: Term	1	0.5144	0.5177	1.0064

Our final decision tree once again showed that the model was not performing any better than its predecessors, with over 20,000 false positives still being recorded.

Event Classification Table

Data Role=TRAIN Target=Loan_Status Target Label=Loan Status

False Negative	True Negative	False Positive	True Positive
0	0	14028	48478

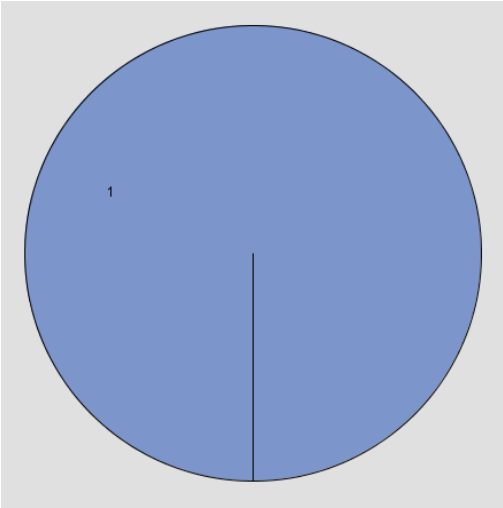
Data Role=VALIDATE Target=Loan_Status Target Label=Loan Status

False Negative	True Negative	False Positive	True Positive
0	0	6012	20777

Cluster Analysis

The fifth model we ran was a cluster analysis. We chose to run a cluster analysis as we thought it may be useful to gain some insight into the clusters that SAS is picking up on within the data, and if we could use that information to predict the class of data. After running the node, which utilized the default settings, and clustering method “Ward”, we found that only 2 clusters had been created, with cluster 1 containing 62,376 of the 62,506 records included in the clustering. Interestingly, regardless of if we changed the clustering method or minimum number of clusters acceptable, the results were identical.

Clustering Criterion	Maximum Relative Change in Cluster Seeds	Improvement in Clustering Criterion	Segment Id	Frequency of Cluster	Root-Mean-Square Standard Deviation	Maximum Distance from Cluster Seed	Nearest Cluster	Distance to Nearest Cluster
0.642657	0.011478	.	1	62376	0.616279	10.17421	2	13.37213
0.642657	0.011478	.	2	130	1.360589	24.91556	1	13.37213



We found from the segment profile node, that the clusters were primarily impacted by one variable, the “Imputed: Maximum Open Credit”. This variable had a worth of 0.003912407 in both clusters, followed by “Imputed: Current Credit Balance”, which lagged behind with a worth of only 0.000255545.

Variable: _SEGMENT_ Segment: 1 Count: 62376
Decision Tree Importance Profiles

Variable	Worth	Rank
IMP_Maximum_Open_Credit	.003912407	1
IMP_Current_Credit_Balance	.000255545	2
IMP_Credit_Score	.000023589	3
IMP_Years_of_Credit_History	.000010379	4
IMP_Annual_Income	.000006694	5
IMP_Monthly_Debt	.000005563	6
IMP_Home_Ownership	.000004011	7
IMP_Purpose	.000003376	8
IMP_Number_of_Open_Accounts	.000001782	9
IMP_Years_in_current_job	.000001464	10

Variable: _SEGMENT_ Segment: 2 Count: 130
Decision Tree Importance Profiles

Variable	Worth	Rank
IMP_Maximum_Open_Credit	.003912407	1
IMP_Current_Credit_Balance	.000255545	2
IMP_Credit_Score	.000023589	3
IMP_Years_of_Credit_History	.000010379	4
IMP_Annual_Income	.000006694	5
IMP_Monthly_Debt	.000005563	6
IMP_Home_Ownership	.000004011	7
IMP_Purpose	.000003376	8
IMP_Number_of_Open_Accounts	.000001782	9
IMP_Years_in_current_job	.000001464	10

Auto Neural Network and Variable Selection

The final model that we explored during this project was the Auto Neural Network. To perform this model, we first used the "Variable Selection" node to reduce the number of variables being used in the neural network, therefore reducing the computational time and effort required to run the model. The variable selection evaluated the variables based upon their R^2 value, and removed any variable that did not contribute to the overall predictive power.

Variable Name	Role	Measurement Level	Type
IMP_Annual_Income	Input	Interval	Numeric
IMP_Bankruptcies	Rejected	Interval	Numeric
IMP_Credit_Score	Rejected	Interval	Numeric
IMP_Current_Credit_Balance	Rejected	Interval	Numeric
IMP_Current_Loan_Amount	Input	Interval	Numeric
IMP_Home_Ownership	Rejected	Nominal	Character
IMP_Maximum_Open_Credit	Rejected	Interval	Numeric
IMP_Monthly_Debt	Rejected	Interval	Numeric
IMP_Number_of_Credit_Problems	Rejected	Interval	Numeric
IMP_Number_of_Open_Accounts	Rejected	Interval	Numeric
IMP_Purpose	Rejected	Nominal	Character
IMP_Tax_Liens	Rejected	Interval	Numeric
IMP_Term	Input	Nominal	Character
IMP_Years_in_current_job	Rejected	Nominal	Character
IMP_Years_of_Credit_History	Rejected	Interval	Numeric

We elected to use the Auto Neural Network so as to allow SAS to examine the different possible models and select the best one for our data. The only setting we changed was the number of hidden units, increasing this from 2 to 3.

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation
Loan_Status	Loan Status	_DFT_	Total Degrees of Free...	62505	.
Loan_Status	Loan Status	_DFE_	Degrees of Freedom f...	62501	.
Loan_Status	Loan Status	_DFM_	Model Degrees of Fre...	4	.
Loan_Status	Loan Status	_NW_	Number of Estimated ...	4	.
Loan_Status	Loan Status	_AIC_	Akaike's Information C...	66568.1	.
Loan_Status	Loan Status	_SBC_	Schwarz's Bayesian C...	66604.27	.
Loan_Status	Loan Status	_ASE_	Average Squared Error	0.174052	0.174056
Loan_Status	Loan Status	_MAX_	Maximum Absolute Err...	0.775586	0.775586
Loan_Status	Loan Status	_DIV_	Divisor for ASE	125010	53578
Loan_Status	Loan Status	_NOBS_	Sum of Frequencies	62505	26789
Loan_Status	Loan Status	_RASE_	Root Average Squared...	0.417196	0.4172
Loan_Status	Loan Status	_SSE_	Sum of Squared Errors	21758.29	9325.568
Loan_Status	Loan Status	_SUMW_	Sum of Case Weights ...	125010	53578
Loan_Status	Loan Status	_FPE_	Final Prediction Error	0.174075	.
Loan_Status	Loan Status	_MSE_	Mean Squared Error	0.174064	0.174056
Loan_Status	Loan Status	_RFPE_	Root Final Prediction ...	0.417223	.
Loan_Status	Loan Status	_RMSE_	Root Mean Squared E...	0.417209	0.4172
Loan_Status	Loan Status	_AVERR_	Average Error Function	0.532438	0.532446
Loan_Status	Loan Status	_ERR_	Error Function	66560.1	28527.4
Loan_Status	Loan Status	_MISC_	Misclassification Rate	0.224414	0.22442
Loan_Status	Loan Status	_WRONG_	Number of Wrong Cla...	14027	6012

The statistics window of the output for our neural network showed that the model had a Misclassification Rate of 22.4414% and 22.442% for the training and validation set, respectively.

In addition, the model had an Average Squared Error of 17.4052% and 17.4056%. In addition, as can be seen from the classification table, the neural network suffered from the same false positive rate as the other models.

Event Classification Table

Data Role=TRAIN Target=Loan_Status Target Label=Loan Status

False Negative	True Negative	False Positive	True Positive
0	0	14028	48478

Data Role=VALIDATE Target=Loan_Status Target Label=Loan Status

False Negative	True Negative	False Positive	True Positive
0	0	6012	20777

Model Comparison and Findings

After running and evaluating the six models we have presented in this report, we finally reached a determination as to the best model of the group. We did this as we mentioned earlier by utilizing the model comparison node, wherein SAS compares key selection statistics, in this case, the validation Misclassification Rate, and determines which model performed best while also ranking the models in order of their performance. From the fit statistics the model comparison provided to us, we can conclude that the best model in the group was the Logistic Regression we conducted first. This model had a Misclassification Rate of 22.4346%. An interesting note is that the first and third decision tree run, as well as the neural network all had a Misclassification Rate of 22.442%, however, we can see that the order in which the models performed is Tree 1, Tree 3, and then the Neural Network. The model that performed worst was the second decision tree, which had a validation Misclassification Rate of 22.4458%, which is just barely higher than the other models, but enough to suggest that it will perform slightly worse.

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Valid: Misclassification Rate
Y	Reg2	Reg2	Logistic Re...	Loan_Status	Loan Status	0.224346
	Tree	Tree	Dec. Tree 2...	Loan_Status	Loan Status	0.22442
	Tree3	Tree3	Dec. Tree 3...	Loan_Status	Loan Status	0.22442
	AutoNeural	AutoNeural	AutoNeural	Loan_Status	Loan Status	0.22442
	Tree2	Tree2	Dec. Tree 3...	Loan_Status	Loan Status	0.224458

Fit Statistics
Model Selection based on Valid: Misclassification Rate (_VMISC_)

Selected Model	Model Node	Model Description	Valid: Misclassification Rate	Train: Average Squared Error	Train: Misclassification Rate	Valid: Average Squared Error
Y	Reg2	Logistic Regression	0.22435	0.16263	0.22438	0.16305
	Tree	Dec. Tree 2 Branch, 6 Depth	0.22442	0.16194	0.22441	0.16293
	Tree3	Dec. Tree 3 Branch, 3 Depth	0.22442	0.16384	0.22441	0.16451
	AutoNeural	AutoNeural	0.22442	0.17405	0.22441	0.17406
	Tree2	Dec. Tree 3 Branch, 6 Depth	0.22446	0.16091	0.22416	0.16229

Now, the purpose of this project was to explore these models, and ultimately, determine if they would provide greater analytical and predictive power than simply using the baseline statistics. In the case of our loan data, the baseline statistics we are comparing the models to are the distributions of the class variables. The distribution table below was previously shown in this report, and provides us with our evaluation statistics for the models. For the loan status “Charged Off”, 22.6383% of the data was contained in this class label, whereas 77.3607% of the data had the class label “Fully Paid”. Therefore, if a firm were to give loans based off of the preliminary base data, the misclassification rate would be 22.6383%.

Distribution of Class Target and Segment Variables
(maximum 500 observations printed)

Data Role=TRAIN

Data Role	Variable Name	Role	Level	Frequency Count	Percent
TRAIN	Loan_Status	TARGET	Fully Paid	77353	77.3607
TRAIN	Loan_Status	TARGET	Charged Off	22636	22.6383
TRAIN	Loan_Status	TARGET		1	0.0010

As we have just discussed, the model comparison node we utilized in SAS EM found that the Logistic Regression model was the best performing model that we analyzed, with a Misclassification Rate for the validation set of 22.4346%. With this, we can see that this model, and in fact all of the models examined, surpassed the baseline Misclassification Rate by approximately 0.2%, indicating that although a small difference, these models, particularly the logistic regression model, should be utilized over simply examining the baseline statistic.

During our analysis, we considered implementing a Principal Component Analysis node prior to running our models, but ultimately, the results did not show any meaningful differences, as the Misclassification Rates for the same models were nearly identical, if not slightly worse, when compared with our existing models, and as such, we decided not to include the Principal Component Analysis as part of our final analysis.

Limitations and Future Analysis

Throughout this report, we have referenced on numerous occasions the issues caused by the missing values and unequal distribution of our target variable classes, “Fully Paid” and “Charged Off”. During our explanation of the models we examined, we mentioned that all models were properly identifying loans as “Fully Paid”, however, the models were performing poorly when it came to properly identifying loans as “Charged Off”, with almost 100% of the

records with this label across all six models being identified incorrectly as “Fully Paid”, creating the approximately 22% Misclassification Rate seen across all of our models.

One potential solution to this problem that we were unable to implement in our analysis is a technique known as oversampling. In this technique, according to a support paper published by the SAS Institute, a sampling method is used which draws an increased number of records from the class label that is considered to be a “rare event”. With this, a more equal number of each class label is selected for use in the analysis, therefore, allowing the models to train and validate on data that is equally distributed between the classes. With this, the models can gain more insight into the variables that affect the classes, enabling them to better address new data and properly classify or group those new instances into the existing data.

If we were to expand upon this project in the future we would likely seek to implement this technique, or one similar, with the goal of improving our models further. With the creation of better models as a result of this oversampling, these models would likely have lower Misclassification Rates, indicating that they are better predictors of the target variable that we are seeking to identify, loan status.

Managerial Implications of this Analysis

We have just discussed the performance of our six models, and the slight improvement upon the baseline statistics that these models provide in terms of their Misclassification Rates, as well as a future technique implementation that may improve the models further. For banks determining who to give loans to, and for how much, these models should be utilized to provide greater insight into the potential for the loans being fully paid off, or for the bank having to charge off the loan. This will allow banks to make better decisions regarding their financial assets, ensuring greater stability and sustainability for the financial sector and economy. We have

seen how with our limited data and examination, these analytical tools have provided models that improve the Misclassification Rate and provide us with better performing models than previously used, albeit very slightly. With further adjustments, such as the oversampling technique just mentioned, these models have the potential to increase in predictive power, and provide even better results than they already have.

References

Arafa, A. (2020, August 8). Bank Loans. Retrieved March 2022, from https://www.kaggle.com/code/abdelrahmanarafa/bankloans34.

Wang, R., Lee, N., & Wei, Y. (n.d.). A Case Study: Improve Classification of Rare Events with SAS Enterprise Miner. SAS Support Papers. Retrieved April 2022, from https://support.sas.com/resources/papers/proceedings15/3282-2015.pdf