Network Analysis of Loan Data

Kevin Sluss

Management 8803 – Spring 2020

4/29/2020

1.0. Introduction

In any large data set there are going to be anomalies. These are caused by human errors, mapping between source systems, or ghosts in the machine. The business world moves fast, and it is not realistic to confirm your data is good. Instead, we institute controls in the data generation process and we rely on these controls and the evidence they are effective. The existence of controls does not relieve the data user from responsibility for making sure the data is fit for use. However, it does allow us to focus our attention on ensuring the data is not definitely bad. In order to assess the quality of the data, we develop predefined rules, with ex. ante defined thresholds. When one of these thresholds is violated, we need a way to quickly identify the root cause of the violation.

In this paper we introduce the idea of using network analysis concepts to target our search for root causes. By visualizing the data set as a population of individuals, we are able to analyze the commonalities between the bad data records and target our search for root causes. This type of methodology has wide ranging applications, for example, into the root cause of diseases. But, in this paper we will focus on using the methodology to analyze the bad data in a loan population at a major bank.

2.0. Background

2.1. Current Expected Credit Loss (CECL) Methodology

In June of 2016, the Financial Accounting Standards Board (FASB) issued a new credit loss accounting standard, which introduced the Current Expected Credit Loss (CECL) methodology for estimating allowances for credit losses. The Bank was required to adopt this methodology for all reporting periods after December 15, 2019. This methodology requires the Bank to estimate the present value of the expected credit loss for each loan in our portfolio over the entire life of the loan. This requires the forecasting of cash flows, determining the probability the cash flow is received, and estimating the recovery rate on the outstanding principal if the cash flow is not received.

2.2. Loan Data

The Bank’s residential portfolio consists of 542,155 loans, broken up between Mortgages, Home Equity Lines of Credit (HELOCs), and Home Equity Installment Loans (HEILs). These loans are originated by multiple lines of business, in different offices throughout the country, utilizing different source systems of record. This leads to variability in the quality of the data and in the definition specific data attributes collected. The loan data fields from each source system are mapped into a common set of variable names, and are then aggregated up into an enterprise data warehouse (EDW). The variables needed to run the loss models are then identified and a final data set is published which contains 94 loan attributes for each of the loans in the portfolio. The final data set is a matrix with 542,155 rows and 94 columns.

The data set which was used for this report is actual February month end production data. Not only is the actual data set much too large for me to upload to Canvas, but, it would also be a violation of my bank's policies for me to share this data set externally. The concepts illustrated in this paper are valid regardless of the number of loans or the number of characteristics. Therefore, for the purposes of this paper, I have created a data set which is analogous to the actual data, but which can be uploaded to Canvas and does not put my career at risk. For example, actual loan numbers were exchanged for sequential numbers and actual property zip codes with a mapping to fictional zip codes. Additionally, the full set of 94 loan characteristics was pared down to a set of 9 characteristics which illustrate the various cases set forth in this paper. None of these manipulations in any way alter the integrity of the results.

3.0. Problem Statement

In a loan portfolio as large and as complex as the one in question, there are likely to be variations and anomalies in the loan origination data. This variability is a natural consequence of multiple lines of business, utilizing different source systems, with data entry being performed by different individuals, in offices scattered throughout the country. Even with standardized processes, anomalies and errors are inevitable. To compound the issue, some of the loans in the portfolio were originated by third parties and purchased by the bank, leading to potentially erroneous or missing data or faulty mapping to our standardized variable names. An example of this would be a customer's current FICO score at the time the loan was purchased being mapped to the origination FICO field in our database.

In a standard CECL execution, the execution team has a matter of days (if not hours) to analyze the data and determine whether or not the data is fit for use. In order to make this problem tractable, the execution team defines a set of rules to which the data must conform. When the data quality rules are violated, the execution team must quickly determine the reason for the violation and whether or not the violation is material.

4.0. Visualizing Loans as a Network

The set of loans in the population is akin to a population of individuals, each with its own set of 94 characteristics. As with a population of individuals, the loans in the population may display similar characteristics based on their backgrounds. For instance, loans from a particular zip code may tend to have a similar market value since market values tend to be reasonably stable in small geographic regions. Similarly, loans originated with an 80% Loan to Value (LTV) may tend to have borrowers with higher FICO scores, as a larger down payment is likely correlated with responsible financial management. The population of loans is similar to a population of individuals, and the 94 characteristics are all potential areas of connection or commonalities between the individuals. Once we identify the bad or anomalous individuals in the population, we can use the characteristics of the population to identify the root cause of the anomalous behavior.

Formally, we have a population of N loans, of which we know that N' exhibit anomalous behavior. If each of the loans in the population has M characteristics, then for every $0 < k \leq M$ of our characteristics we get an NxN adjacency matrix A_k. For every loan $i \leq N$ we have a row with N columns, the entries in that row corresponding to the values of 1 or 0 depending on whether or not loan $j \leq M$ has the same value as loan i for characteristic k.

$$A_{k_{i,j}} = \begin{cases} 0\ if\ loan\ i\ and\ loan\ j\ have\ different\ values\ for\ characteristic\ k \\ 1\ if\ loan\ i\ and\ loan\ j\ have\ the\ same\ value\ for\ characteristic\ k \end{cases}$$

For each characteristic there will be sets of components. For example, in the zip code matrix, all loans in a particular zip code will make up a component. Once we have the adjacency matrix, and we have determined all of the components for a particular adjacency matrix, we look for a component that is largely made up of members of our bad data set. Figure 1 shows a visual representation of an important an important variable.

Figure 1 Example of an Important Variable

$$A_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & \mathbf{1} \\ 0 & 0 & 0 & \mathbf{1} & \mathbf{1} \end{bmatrix} \quad In = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix}$$

For this variable, we see that there is a component in the variable k adjacency matrix which is entirely made up of loans from our bad data set. This does not tell us what is the reason for the bad data, but we do know that the loans in the bad data set share a common characteristic for variable k. An example of this would be a set of bad data loans which all come from the same source system. This tells us that, whatever is the source system, it likely contributes to the fact that they are bad. Perhaps they all come from a line of business with poor data quality controls. Or, maybe there is a problem mapping the variable name for variable k from that particular source system to the common name. Whatever is the case, we know that variable k is where we should look for the reason behind the bad data.

Figure 2 shows an example of a not important variable. In this figure we see that there is a component, and that component does contain one of our bad loans, but the component is not substantially made up of loans from our bad data set. This is similar to the situation where there are some bad loans in a particular zip code, but the fact they are in the same zip code does not contribute to the fact that they are bad.

Figure 2 Example of a Not Important Variable

$$A_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad In = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix}$$

5.0. Real World Application

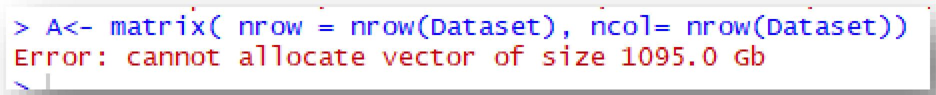
5.1. Generating the Edges for Each Variable Graph

Because we cannot monitor every variable, we setup data quality rules which can key us in to potential errors in key data elements (KDEs). When one of these KDE data quality rules is violated, we must investigate it and determine whether or not it is a material issue which needs to be addressed. If we determine it is not a material issue which needs to be addressed through an additional overlay to our modeled reserve, we must have a very good explanation why we came to that conclusion. This explanation needs to be supported by analysis, with evidence retained for future review by Internal Audit, the Federal Reserve, the Office of the Comptroller of the Currency, and any other oversight entity which knocks on our door. In general, we have 24 hours to analyze the data, identify potential areas of concern, investigate the areas of concern, and determine whether or not to proceed with the execution.

One of our KDEs is the variable CURRENT_DELINQUENCY_STATUS. This variable is a key indicator of potential default, as it is unlikely (maybe impossible) for a loan to transition from 0 days delinquent to default. In reality, the probability of default increases significantly as the delinquency status of the loan increases. There are 5 acceptable values for this variable. The acceptable values are 0, 30, 60, 90, 120, 150, and 180. One of the data quality rules which we setup for this variable is the number of instances where it is missing. The threshold for missing values is 1% of the loan population. If this value is missing for more than 1% of the loan population, we must investigate and determine whether or not we need to add an additional measure of conservatism to our modeled output. In the second quarter of 2020, this value was missing in 1.13% of the loans in the population. In order to illustrate the principals discussed in this paper, we will attempt to determine the reason for this bad data. In this particular situation, we already know the reason for the bad data. However, this does not diminish the findings of this paper. As discussed earlier, utilizing a data set for which we know the root cause of the error allows us to whittle down the data set to a size which can be uploaded to Canvas while still successfully demonstrating the methodology. The fact is that, had we had the techniques discussed in the paper available to us, we would have been able to identify the root cause much easier than we were through our brute force approach.

The first step in the process is to create a data set named "Delinq" which contains the loan data for the loans with missing values for CURRENT_DELINQUENCY_STATUS. This data set contains 6,145 loans. Then, for each of our characteristics[1] k, we generate the adjacency matrix A_k. This is actually harder than it sounds as attempting to create a matrix with 542,155 rows and columns results in a matrix of over 1TB in size. This is beyond my computer's capacity to achieve as shown in Figure 3.

Figure 3 Very Large Matrix Error



```
> A<- matrix( nrow = nrow(Dataset), ncol= nrow(Dataset))
Error: cannot allocate vector of size 1095.0 Gb
>
```

In order to circumvent this problem, we utilized loops to determine the edges for each of our adjacency matrices. We create two data frames "From" and "To" which will house the From and To nodes for each of our edges. However, the size of the dataset again presents a problem. The total possible edges for each of the variables in our data set is in our data set is 146,965,750,935. Brute force analysis would mean that we would have to check each of these edges for each of our variables. This is not realistic. Furthermore, a completely connected graph for a given variable would have 146,965,750,935 edges.

[1] We did not develop an adjacency matrix for the field UNQ_ID_IN_SRC_SYST as this field is unique to each loan and cannot tell us anything about the relationship to the loan and other loans.

Which is beyond R's ability to process, even on my high powered laptop. Instead we utilize the fact that, if a set of loans all have a particular value for a given variable, then they don't have any other value for that variable, thus, they make up a component and they don't need to be checked for agreement with other loans. i.e. if loans 1, 2, 3 have the same the value for variable k then it is sufficient to state that 1→2 and 1→3. We can rely on commutability and we do not need to list all of the edges. Therefore, for each of our variables, we apply the following algorithm to generate a list of edges:

1. Create a dataset named TempData which is equal to the original data set.
2. Start with loan 1.
3. Create a data frame "Temp" which contains the loan IDs of all the loans which share the same value as loan 1.
4. Determine the number of loans, n, in the Temp data set.
5. In the From matrix, repeat the first loan ID in the TempData dataset n times .
6. In the To matrix, enter the Temp data set.
7. Remove all the loans in the Temp data set from the TempData data set.
8. Repeat until there are no loans in the TempData set.

Using this algorithm, the number of iterations needed to determine all the edges for each variable is equal to the number of unique values for the particular variable. We can then use the simplify function from the igraph package along with the graph_from_data_frame function to generate a graph of the edges for each variable.

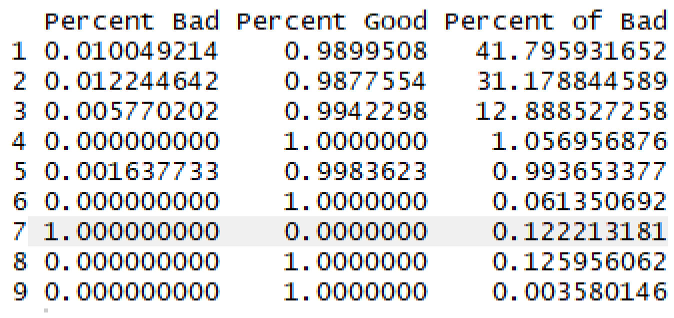
5.2. Determining the Importance of the Variable

In order to determine the importance of each variable there are two considerations.

1. What percentage of each component is made up of loans from our bad data set?
2. What percentage of our bad data set falls into components which aren't largely made up of loans from our bad data set?

The perfect situation is a variable for which there is a set of components, all of which are 100% made up of loans from our bad data set, and which encompass the entirety of our bad data set. Using the components function in R we are able to get the component membership of each loan in our data set. We then determine what percentage of each component is made up of loans from our bad data set and what percentage of each component is made up of loans not from our bad data set. Table 1 shows an example of this for the variable AMORT_TYPE. The Percent Bad columns shows what percentage of each component is made up of bad data. The Percent of Bad column shows what percentage of our bad data set falls into each component. From this table we see that component 7 is 100% made up of loans from the bad data set. However, this component only contains 12% of our bad data set. Furthermore, there are no components which are largely made up of bad data. This indicates that, while there is some correlation between component 7 of AMORT_TYPE and bad data, there is not likely to be causation between any aspect of the AMORT_TYPE variable and our bad data.

Table 1 Component Makeup for AMORT_TYPE



```
  Percent Bad Percent Good Percent of Bad
1 0.010049214    0.9899508   41.795931652
2 0.012244642    0.9877554   31.178844589
3 0.005770202    0.9942298   12.888527258
4 0.000000000    1.0000000    1.056956876
5 0.001637733    0.9983623    0.993653377
6 0.000000000    1.0000000    0.061350692
7 1.000000000    0.0000000    0.122213181
8 0.000000000    1.0000000    0.125956062
9 0.000000000    1.0000000    0.003580146
```

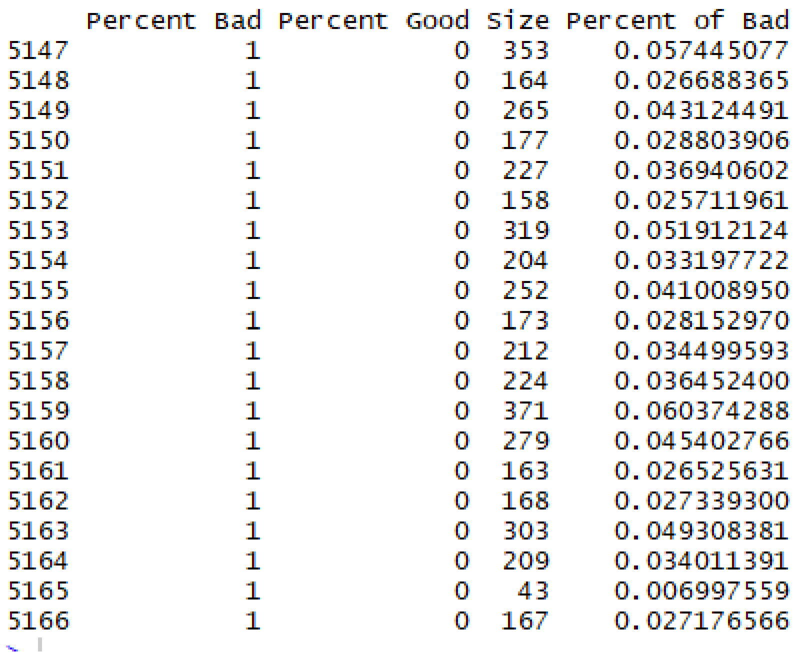
The AMORT_TYPE variable is easy to visually investigate because it is a categorical variable there are only nine components. The problem becomes much more complex when we analyze a variable with many more components, such as "ORIGINATION_DATE", which has 10,210 components. In order to analyze these more diverse variables, we need to have a summary statistic by which we can rank their relative importance. For this purpose, we define a threshold for the percentage of the component made up of bad data loans ("Percent Bad" from Table 1) and we define a data set made up entirely of the components for which with the percentage of bad loans is greater than our threshold. We then define the variable Score as the Percent Bad times the relative size of the component in relation to our bad data set ("Percentage of Bad" from Table 1).

$$Score = Percent\ Bad * Percentage\ of\ Bad$$

This provides us a single scoring number by which we can rank the importance of the variable.

As an example, suppose we set Threshold = 1. The AMORT_TYPE variable would have a score of 12.2%, as can be seen in Table 1. Meanwhile the "ORIGINATION_DATE" variable would have a score of 72.11%, as can be seen from Table 2.

Table 2 Origination Date - Components with Percent Bad =1



```
     Percent Bad Percent Good Size Percent of Bad
5147           1            0  353    0.057445077
5148           1            0  164    0.026688365
5149           1            0  265    0.043124491
5150           1            0  177    0.028803906
5151           1            0  227    0.036940602
5152           1            0  158    0.025711961
5153           1            0  319    0.051912124
5154           1            0  204    0.033197722
5155           1            0  252    0.041008950
5156           1            0  173    0.028152970
5157           1            0  212    0.034499593
5158           1            0  224    0.036452400
5159           1            0  371    0.060374288
5160           1            0  279    0.045402766
5161           1            0  163    0.026525631
5162           1            0  168    0.027339300
5163           1            0  303    0.049308381
5164           1            0  209    0.034011391
5165           1            0   43    0.006997559
5166           1            0  167    0.027176566
```

Table 3 and Table 4 display the score for each of our variables at thresholds of 1 and .9 respectively. As you can see from the different scores for the FIRST_PAYMENT_DATE variable, there is some need to calibrate the thresholds. And, of course, the CURRENT_DELINQUENCY_STATUS variable has a score of 100%, as would be expected.

Table 3 Scores for Each Variable with Threshold = 1

Variable	Score
AMORT_TYPE	12.22%
BANKRUPTCY_STATUS	0.00%
CURRENT_FICO_SCORE	12.14%
ZIP_CODE	0.00%
FIRST_PAYMENT_DATE	18.32%
ORIGINATION_DATE	72.11%
CURRENT_DELINQUENCY_STATUS	100.00%
CURRENT_LOAN_BALANCE	0.00%

Table 4 Scores for Each Variable with Threshold = 0.9

Variable	Score
AMORT_TYPE	12.22%
BANKRUPTCY_STATUS	0.00%
CURRENT_FICO_SCORE	12.14%

ZIP_CODE	0.00%
FIRST_PAYMENT_DATE	56.60%
ORIGINATION_DATE	85.09%
CURRENT_DELINQUENCY_STATUS	100.00%
CURRENT_LOAN_BALANCE	0.00%

5.3. Using Variable Scores Target the Investigation

From the scores generated in the previous section we see that the ORIGINATION_DATE and FIRST_PAYMENT_DATE variables are important variable. Figure 4 and Figure 5 show the histograms of ORIGINATION_DATE and FIRST_PAYMENT_DATE respectively for the loans with missing values for CURRENT_DELINQUENCY_STATUS. From this we can see that a large number of loans in our bad data loans are originations after 1/27/2020, and with first payment dates after 3/1/2020. Because our data set was snapshot as of 2/29/2020, these loans have not had an opportunity to make a first payment. Therefore, they cannot be delinquent.

Figure 4 Histogram of ORIGINATION_DATE

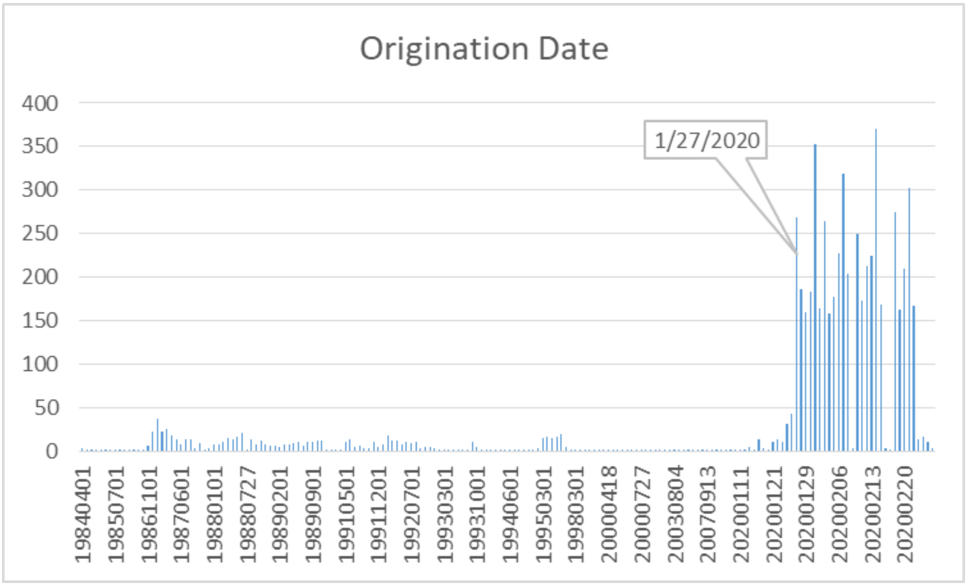
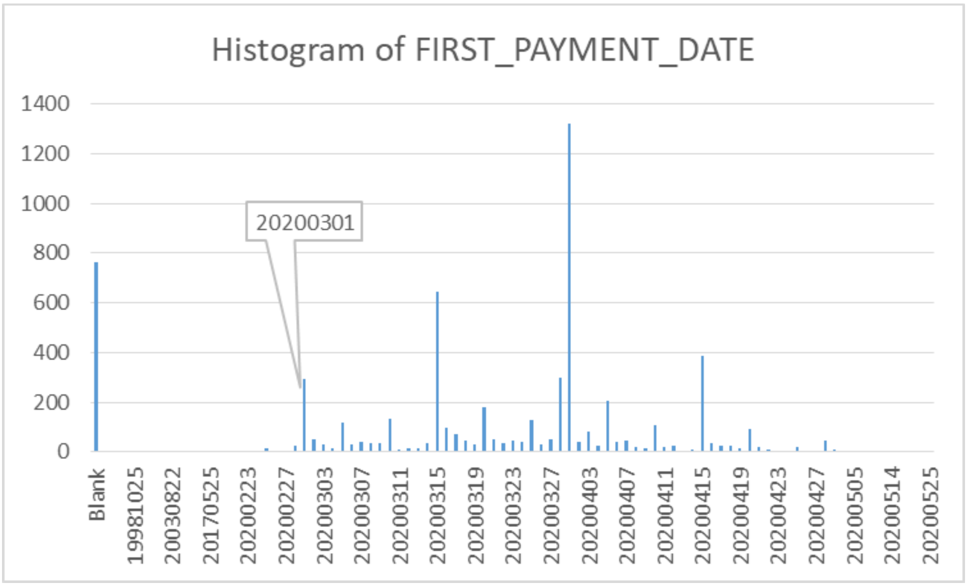


Figure 5 Histogram of FIRST_PAYMENT_DATE



After eliminating the 5,316 loans with a first payment due after 3/1/2020, we are left with 829 questionable loans, which constitutes 0.15% of our total data set. A large number of these questionable loans do appear to be problematic and will need to be investigated. However, given the time constraints of the CECL execution, it is not realistic to investigate the problem during the run. The more important conclusion is whether or not the bad data represents a material deficiency in our reserving process which would need to be remediated through the use of a model overlay. This decision is beyond the scope of this paper.

6.0. Conclusion

In this paper we have demonstrated how the concepts of network analysis can be used to investigate anomalies in a large and complex data set. The purpose of this methodology is not to identify the root cause of the issue. Rather, it is to allow us to quickly identify where we should look for the root cause. Rather than investigating 94 variables, we are able to quickly zero in on the commonalities in our bad data set, and limit our investigation to those variables. Some tuning of the scoring threshold is required, and, there are significant real world obstacles to the deployment of this methodology, primarily relating to the use of R in the analysis of large data sets. Fortunately, in the business world there are also real world solutions to these problems. For example, when we deploy this solution in our business we will be able to harness the Bank's grid computing solutions, which are specifically designed for the analysis of big data. This will require us to rebuild the programs presented here in Python or SAS, but that is a small obstacle. What we have shown here is that, by viewing the large data set as a population with characteristics in common, we can quickly isolate commonalities and target our investigation into the root cause of the issue.