University of Connecticut

MS in Business Analytics and Project Management

OPIM-5671- Data Mining and Business Intelligence

Professor Cruz

**Text Mining Project**

**Consumer Complaints Dataset**

Group 4:

Abhinav Dubey

Jiaxuan Wang

Marina Suberlyak

Shaista Usman

**Table of Contents**

[**Problem Statement**](#_heading=h.g72yppld15os) **2**

[**Methodology**](#_heading=h.yf0qqae9okq5) **2**

[**Sample**](#_heading=h.8uwhpo8mslgl) **2**

[**Explore**](#_heading=h.bkhk65iidb5c) **3**

[**Modify**](#_heading=h.xjuqdxxypm51) **6**

[Filter Node](#_heading=h.kyju12cj5hfk) 6

[Sample and Partition Node](#_heading=h.u1h05sr8gzr4) 6

[SAS Code Node](#_heading=h.105dap6tvmg3) 7

[Text Parsing Node](#_heading=h.u0dqrq7pjzi3) 8

[Text Filter Node](#_heading=h.8p44d9ah5iu3) 8

[Text Cluster Node](#_heading=h.x1khmcr76dpj) 9

[Text Topic Node](#_heading=h.seovu2fydbqt) 10

[Text Profile Node](#_heading=h.u3e2x9aahict) 11

[Metadata Node](#_heading=h.l8a4m5gyfk7f) 11

[**Model**](#_heading=h.shwm6othwjho) **12**

[Decision Tree Node](#_heading=h.nfs933a7bo0v) 12

[Regression Node](#_heading=h.57rwgvnkr34x) 15

[MBR Node](#_heading=h.m3rji94rjoij) 16

[Neural Network Node](#_heading=h.4z4f2j584zqy) 19

[**Assess**](#_heading=h.antj3obkukz) **21**

[Model Comparison Node](#_heading=h.ucriwp661wdk) 21

[**Results**](#_heading=h.3yo3w2eth9zn) **23**

[**Conclusions and Recommendations**](#_heading=h.jkdg09mzruf6) **23**

[**References**](#_heading=h.a5x9hd5s2tx8) **24**

## 

# Problem Statement

Our dataset compiles consumer complaints received by the Bureau of Consumer Financial Protection about financial products and services for the years 2021-2016. Complaints are processed by the institutions and each complaint has to identify information about the issue, the resolution or latest status, and whether the consumer filed a dispute to the response provided by the institution. The intention is to predict whether the resolution will be disputed by the consumer from the variables available; as some variables are text, text mining techniques will be applied to derive input values for modeling. We presume the business application here is twofold: inform which products, services, and resolution responses offered by the institution require a review to improve clarity, usability, and value to the consumer; and understand when a consumer is likely to file a dispute to guide the response and training for the institution’s employees processing the claims.

# Methodology

### Sample

The data contains a variety of fields, including company name, issue within the complaint, the products and sub-products for the complaint, company response and timing of the response, and identifying data such as dates, state, and zip code. There are 670,598 rows and 15 columns in the dataset, providing sufficient predictor information to build our models. All columns are listed below:

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **NAME** | **DESCRIPTION** | **Data type** |
| 1 | Company | Company names like Wells Fargo & Company, Citibank, Bank of America, etc | Nominal |
| 2 | Company\_response\_to\_consumer | Company response to the customer | Nominal |
| 3 | Complaint\_ID | Complaint ID | ID |
| 4 | Consumer\_consent\_provided\_ | Consumer consent provided | Nominal |
| 5 | Consumer\_disputed\_ | If a consumer disputed or not | Binary |
| 6 | Date\_received | The date on which the consumer files the complaint | Interval |
| 7 | Date\_sent\_to\_company | The date on which complaint is sent to the company | Interval |
| 8 | Issue | Issues for which consumer has filed a complaint | Text |
| 9 | Product | Product for which complaint is filed | Nominal |
| 10 | State | Geographic identifier to know in which state complaint is filed like CT, VA, MA, etc | Nominal |
| 11 | Sub\_product | Specific sub-product of the product to which the consumer files the complaint. | Text |
| 12 | Submitted\_via | The mode used by the consumer to file the complaint. | Nominal |
| 13 | Tags | Gives extra information about the consumer | Nominal |
| 14 | Timely\_response\_ | Tell if the response to the complaint was on time or not | Binary |
| 15 | ZIP\_code | Geographic identifier to know where the complaint was registered. | Numeric |

### Explore

During the review of the data, the following decisions were made to reduce the dataset to its most useful form.

1. Company**:** This variable indicates a unique name of the company receiving the complaint and in itself holds no information for prediction. However, the company name may be useful for tabulation once we complete the analysis.
2. Consumer\_consent\_provided**:** This variable has “n/a” for over 60% of the data. Without clarity on why “n/a” might be an acceptable value, we choose to reject the variable.

|  |
| --- |
|  |

1. Date\_received and Date\_sent\_to\_company**:** These variables contain the time id of the complaint. We hypothesize that time separation between the consumer submitting the complaint and company receiving the complaint has no relationship to the type of complaint and subsequent resolution steps and is therefore not relevant in predicting consumer likelihood to dispute the resolution.
2. State**:** This variable is a geographic identifier and on visual inspection, the consumer dispute variable is equally represented across all states. Due to the complexity of incorporating geo data into modeling, we choose to reject this analysis.

|  |
| --- |
|  |

1. Tags**:** This variable is dominated by blank values and is therefore not valuable as a predictor.

|  |
| --- |
|  |

1. Timely\_response: This variable is dominated by a single value, “Yes”, and is therefore not useful in this analysis.

|  |
| --- |
|  |

1. Zipcode**:** This variable is another geographic identifier and requires additional context in order to become useful. We hypothesize there is not a meaningful relationship between consumer likelihood to dispute and geography and therefore reject zipcode.

Please find the screenshot below of the metadata after rejecting the above variables.

|  |
| --- |
|  |

We now take a look at two text variables we identified - issue and sub-product - to understand how diverse and complex the data is within. In all, variable issue has 92 levels and a handful of categories contain a lot of the frequency; at the same time, target variables appear distributed across all levels. Given that out of 15K observations, we discover 92 unique categories, we hypothesize that the issue contains some level of discrete answers, which were predetermined by the institution and made available for the consumer to select during complaint filing.

|  |
| --- |
|  |

Variable sub-product has 48 levels and also has a handful of categories with a lot of frequency.

|  |
| --- |
|  |

### Modify

#### Filter Node

We identify that the sub-product variable has ~30% values missing and we cannot perform imputation without industry knowledge or additional data. Instead, with the dataset of 670K observation, we can afford to remove the missing rows without compromising the overall information. We apply a Filter node to achieve this.

|  |
| --- |
|  |

#### Sample and Partition Node

The reduced dataset includes 472K observations. To make it more manageable in modeling, we extract a random stratified sample of 15,000 observations, using the below setup:

|  |  |
| --- | --- |
| Sample method: stratify  Type: number of observations  Observations: 15,000  Stratified criterion: proportional to preserve the proportions from original data |  |

Lastly, we partition the sample into train/validate/test to allow model assessment and comparison; we will allocate 60% of the data to training, and split remainder into validation and test where we will compare model performance to confirm no overfitting and no significant deterioration in key indicators. Our setup and confirmation are below:

|  |  |
| --- | --- |
| Training: 60%  Validation: 20%  Test: 20% |  |

#### SAS Code Node

Without a start or stop list provided directly, we chose to perform a preliminary analysis where we combined SAS Code node with Text Parsing and Text Filter nodes to generate start and stop lists against the source variable issue. We relied on frequency filtering to derive the lists to allow us to minimize noise in the terms frequently repeated but without material information.

|  |
| --- |
| **Creating a Start/Stop List** |
|  |

|  |
| --- |
| **SAS Code Node – Code Editor** |
|  |

#### Text Parsing Node

We have previously identified 2 text variables: issue and sub-product. However, variable issue contains twice as many unique categories (92 vs. 48 levels), and we choose this variable to derive additional information.

Applying Text Parsing node, our setup included selecting detection for parts of speech, removing default synonym list, adding the previously derived stop list and selecting English as the language.

|  |
| --- |
| **Property Panel** |
|  |

Completing this node, we were able to decompose text into tokens and identify terms, associate each term with parts of speech and perform stemming to equate terms with different tenses.

#### Text Filter Node

To further improve the quality of terms derived from variable issue, we applied a Text Filter node, which assists in correcting misspellings, establishing frequency and terms weights. Our setup included “Check Spelling” and default settings for frequency and term weighting.

From the initial run, we discover that a single term, “loan”, appears in an extraordinary number of documents. We also observe that term frequency plotted against # of documents is a 45° line, so many terms appear approximately once in many documents.

|  |
| --- |
| **Zipf Plot and Number of Documents by Frequency** |
|  |

We choose to select Default frequency weighting and Inverse Document Frequency term weighting to give more emphasis to infrequent terms in the document collection and produce variables for modeling. The resulting weight to doc # plot is below:

|  |
| --- |
| **Number of Documents By Weight** |
|  |

#### Text Cluster Node

In order to reduce the dimensionality of our dataset and to transform the weighted, term-document frequency matrix generated by the filter node into SVD values, we add a Text Cluster node. After trialing cluster sizes, we found 4 clusters to be optimal for our dataset as they provide adequate distance among them such that each cluster possesses a distinct theme or concept and is mutually exclusive, as evidenced by the distances between each of the four clusters. We do note that cluster size is uneven, with cluster #2 hosting majority of the observations, with terms around account servicing and payment issues. In a dataset around financial product complaints, it makes sense that most of the issues would be associated with consumer’s accounts and their grievances against the financial products or services availed by them. We also note the 2nd largest cluster contains terms around debt collection; again, this makes sense as consumers are likely to have disagreements over such financially impactful judgement.

|  |
| --- |
| **Property Panel, Cluster Frequencies and Distance Between Clusters** |
|  |

|  |
| --- |
| **Text Clusters** |
|  |

#### Text Topic Node

We further added a text topic node in series with the cluster node to refine our term understanding and increase the number of variables derived from text to aid in modeling later. We used all the default values for the topic node. We settle on fifteen topics and observe that only a few terms define each topic. This again supports our hypothesis that the variable issue contains predetermined text that likely already separates issue topics into discrete categories.

We used these distinct topics as inputs to our models for accurate classifications.

|  |
| --- |
| **Text Topics** |
|  |

#### Text Profile Node

Given our observation that text within variable issue is pre-crafted to contain discrete responses, we take a look at the Text Profile node to uncover whether certain terms, and therefore topics, have a relationship with the target variable. As per SAS documentation, “For each level of a target variable, the node outputs a list of terms from the collection that characterize or describe that level.”1

|  |
| --- |
| **Profiled Variables** |
|  |

The output shows that target variable consumer\_disputed=yes commonly has terms around the financial product itself or the application process; whereas consumer\_disputed=no often contains terms around communication, account services and debt collection.

#### Metadata Node

We apply Metadata node to assign roles to the new variables we derived during text analysis steps. Specifically, we will set the first 10 topic variables as inputs, and keep all other variables at default.

Text Import node is not applicable since our dataset is directly provided. Text Rule-Builder is also not applicable since we are relying on predictive models to provide results instead.

### 

### Model

Predictive modeling helps us to find good rules(Models) for guessing the values of one or more variables in a data set from the values of the other variables in the data set. Once a good rule has been found, it can be applied to the new datasets to predict future scenarios. As in the given consumer complaints dataset our target variable “Consumer\_disputed\_” is categorical, which gives us information about the customer if he/she is disputing or not. To predict future values we have done four models: decision tree, regression, MBR and Neural Network.

|  |
| --- |
|  |

#### Decision Tree Node

To do decision tree modeling attach a decision tree node to the Metadata node. Change the Assessment Measure property in the subtree section to Average Square Error. Run the Decision Tree node.

|  |
| --- |
| **Property Panel - Decision Tree Node** |
|  |

Once the model executed successfully after that check the results as our primary goal is to run different models and select one from it, but it might be informative to examine how the decision tree chose to partition the data.

Please find the screenshots of the results which we got from the running the decision tree node.

|  |
| --- |
| **Variable Importance** |
|  |

|  |
| --- |
| **Decision Tree** |
|  |

|  |
| --- |
| **Fit Statistics** |
|  |

|  |
| --- |
| **Classification Chart** |
|  |

|  |
| --- |
| **Cumulative Lift** |
|  |

From the above screenshots we can see in the fit statistics results the model is not overfitting because the difference between the training misclassification rate and validation misclassification rate is not much. And same for the training misclassification rate and test misclassification rate.

In this model, we have identified the most important predictor variable to predict our target. The most important variable is “Sumitted\_via”

#### Regression Node

To do the regression modeling attach a Regression node to the Metadata node. After attaching the node, we have not changed any property panel properties. Run the Regression node.

|  |
| --- |
| **Fit Statistics** |
|  |

|  |
| --- |
| **Classification Chart** |
|  |

|  |
| --- |
| **Cumulative Lift** |
|  |

From the above screenshots we can see in the fit statistics results the regression model is not overfitting because the difference between the training misclassification rate and validation misclassification rate is not much. And same for the training misclassification rate and test misclassification rate.

#### MBR Node

To do MBR modeling attach an MBR (Memory Based Reasoning) node to the Metadata node. Change the Number of Neighbors property in the Train section to 8. Select the Variables property, and change the Use status of all input variables to Rejected. Then change the Use status of all TextCluster\_SVDn variables to

Yes. These variables are orthogonal, and hence can be used as inputs to the MBR node. Otherwise,

you would need to use a method such as principal components to convert inputs to orthogonal inputs.

Run the MBR node.

|  |
| --- |
| **MBR Property Panel** |
|  |

|  |
| --- |
| **Variable Property** |
|  |

|  |
| --- |
| **Fit Statistics** |
|  |

|  |
| --- |
| **Classification Chart** |
|  |

|  |
| --- |
| **Cumulative Lift** |
|  |

From the above screenshots we can see in the fit statistics results the MBR model may be overfitting because the difference between the training misclassification rate and validation misclassification rate is significant. And same for the training misclassification rate and test misclassification rate.

#### Neural Network Node

To do neural network modeling Attach a Neural Network node to the Metadata node. We have not changed any property panel settings. We executed the node to see the results.

|  |
| --- |
| **Fit Statistics** |
|  |

|  |
| --- |
| **Classification Chart** |
|  |

|  |
| --- |
| **Cumulative Lift** |
|  |

From the above screenshots we can see in the fit statistics results the neural model is not overfitting because the difference between the training misclassification rate and validation misclassification rate is not much. And same for the training misclassification rate and test misclassification rate.

### Assess

#### Model Comparison Node

After creating and running our models, we used the Model Comparison node to assess the corresponding results and to determine our best model. We decided on measuring each model’s performance based on fit statistics like misclassification rate, RMSE, and MSE and the ROC and lift charts.

* Misclassification Rate : The percentage of classifications that were incorrect, the values closer to zero are better.
* Mean Square Error : MSE measures the [average](https://en.wikipedia.org/wiki/Expected_value) of the squares of the [errors](https://en.wikipedia.org/wiki/Error_(statistics))—that is, the average squared difference between the estimated values and the actual value. In general, a lower RMSE is better than a higher one.
* Root Mean Square Error : RMSE is the square root of the average of squared errors. In general, a lower RMSE is better than a higher one.
* ROC Chart : Shows the trade-off between sensitivity and specificity. Classifiers that give curves closer to the top-left corner indicate a better performance.

|  |
| --- |
| **Fit Statistics** |
|  |

|  |
| --- |
| **Cumulative Lift** |
|  |

|  |
| --- |
| **ROC Chart** |
|  |

# 

# Results

Based on the results obtained from the ‘Model Comparison’ node, our best model overall was the Decision Tree Model. We made our decision based on the training, validation, and test misclassification rate and RMSE. The Decision Tree model has a training and validation misclassification of about 0.2184 and a validation RMSE of 0.2089 and MSE of 0.168 an overall accuracy of 71.86%. The tree model further has a test MSE of 0.1678 and a misclassification rate of 0.2188 and a RMSE of 0.4096. The model performs better on our test partition without overfitting the training or validation datasets. The tree model further has lower RMSE and misclassification rates when compared to our other models. The tree model provides a lift of 1.14 and 1.21 on validation and test partitions, respectively. The neural network model has a better lift but has similar misclassification rates and RMSE when compared with the tree mode. The tree model further has a ROC index of 0.59 on the test partition which is the same for most of our models and a sizable area under the curve which is indicative of better accuracy and of a stronger model. The tree model captures sufficient variance and inherent patterns from the training partition while remaining relatively less complex and performing well on the validation and test partitions. The tree model has a better rate of classification and a higher accuracy of predictions for both classes of the target variable.

# Conclusions and Recommendations

From our text mining analysis, we can identify Cluster 2 as grouping with the largest amount of absolute complaints; as a reminder, it is also our largest cluster and the terms that define it are [loan, collection, foreclosure, modification, account, +payment, 'escrow account', +service]. This provides immediate learnings for the business as they can examine the terms that typically accompany a dispute and begin to consider what actions they may wish to pursue.

|  |
| --- |
|  |

For modeling, identifying potential disputes correctly in turn can lead to process efficiencies or training opportunities for the business. Therefore, we conclude misclassification rate is important in evaluating the performance and any improvements to the model should aim to minimize this metric, particularly false negatives as those especially indicate missed opportunities for the business.

We see a few paths to improving our model from misclassification rate of 21.88% on Test partition:

1. **More precise stop and synonym lists**: For text parsing and filtering, we derived a stop list from the text variable itself. If the domain experts participate in the stop or start compilation, it is likely more noise can be eliminated. Similarly, a synonym list that accurately mimics industry language and commonly interchangeable terms can be helpful in deriving more value from the text variable.
2. **Refine text variable weights**: As we noted previously, the variable issue appears to have predetermined statements that consumers selected from the list; we expect that these statements already have a certain level of information separation. We believe this impacts how terms should be weighted and consulting the domain experts may be helpful here to further refine this text variable for modeling.

# References

1 SAS Enterprise Miner 14.3 Reference Help

<https://documentation.sas.com/?docsetId=tmref&docsetTarget=n0sebvirmou078n1sxso7lqkwyln.htm&docsetVersion=14.3&locale=en>