Demo with Driver Feedback

Using Text Rule Builder to Categorize Documents

Case Study: Text Analysis of Driver's Feedback (continued)

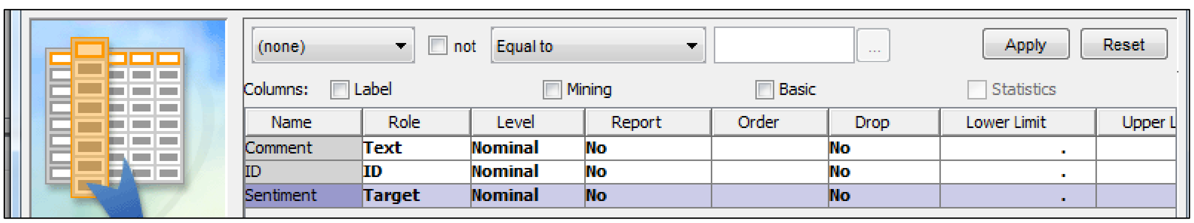
This case study is in my text book. In this part of the case study, you first build a Text Rule Builder model to categorize positive versus negative feedbacks. (These are feedbacks by professional drivers that were classified as positive or negative by company experts.)

Data

To use Text Rule Builder for building models, both positive and negative comments must be combined in the same data set.

- **All_model.sas7bdat** (a data set that combines positive and negative comments for building models. This data set has 90% of all comments.) – This will be used for building models.
- **All_test.sas7bdat** (a data set that combines positive and negative comments for testing models built. This data set has 10% of all comments.) – This will be used for scoring, and because it has the original categorization by experts, it can be used to check the performance of scoring models.
- **Engdict.sas7bdat** (to be used as a dictionary – created from opensource dictionary)

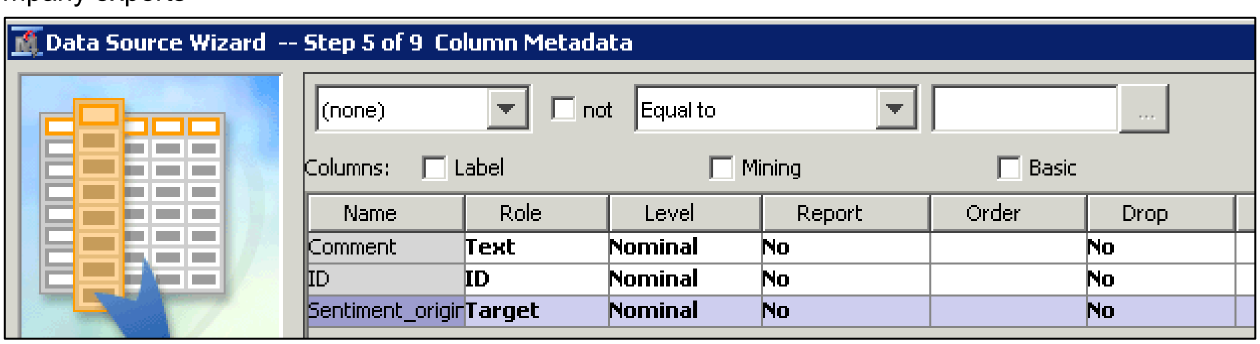
1. Create a new project in SAS Enterprise Miner.
2. Create a new library (name it **Course**) and point it to where the data are located
3. Add the data set **ALL_MODEL** to the project. Use all default options in data creation steps except change the role of the variable **Sentiment** to **Target** as shown below. The **Sentiment** variable reflects whether the comment is judged as positive or negative by company experts. In this demonstration, it is used as a target variable to derive the rules.



Name	Role	Level	Report	Order	Drop	Lower Limit	Upper L
Comment	Text	Nominal	No		No	.	
ID	ID	Nominal	No		No	.	
Sentiment	Target	Nominal	No		No	.	

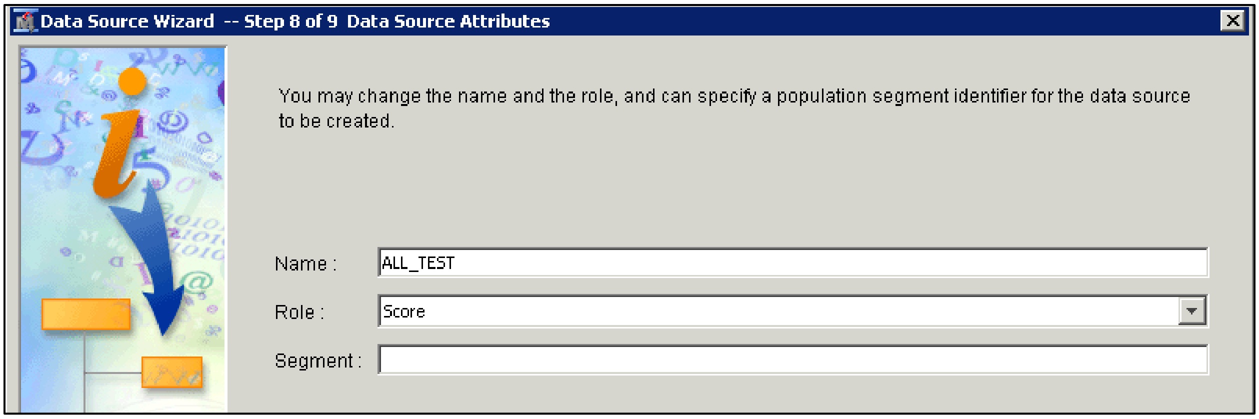
4. Add the data set **ALL_TEST** to the project. Use all default options in the data creation steps except as follow:

   In Step 5, change the role of the variable **Sentiment_original** to **Target**. The **Sentiment_original** variable reflects whether the comment is judged as positive or negative by company experts

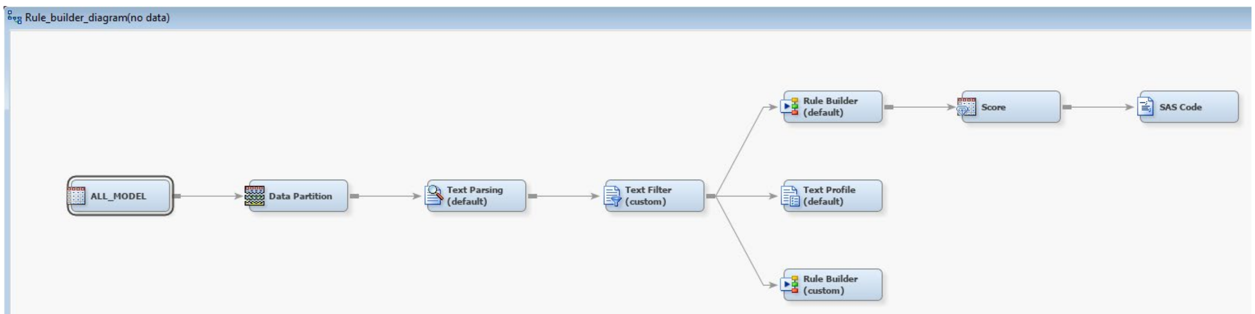


Name	Role	Level	Report	Order	Drop
Comment	Text	Nominal	No		No
ID	ID	Nominal	No		No
Sentiment_origir	Target	Nominal	No		No

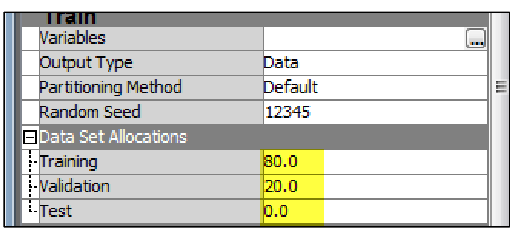
- In Step 8, change the role of the data source to **Score** as shown below.



5. Import the XML diagram titled “**Rule_builder_diagram(no data)**” into your project. Then drag the data set **ALL_MODEL** onto the diagram space. Connect the data set with the **Data Partition** node as shown below.



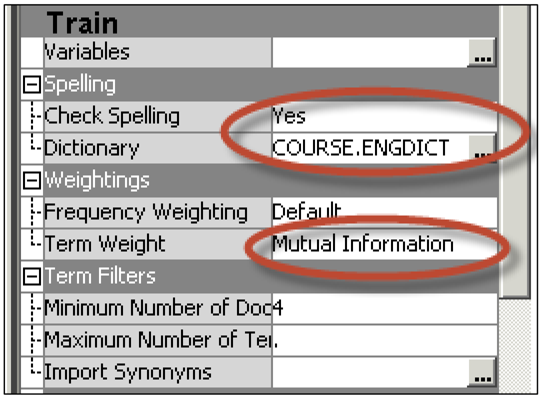
6. Go to the properties panel of the Data Partition node and *note* the value of Data Set Allocations as shown below. Right-click and run the node.



7. Use all default properties in the Text Parsing (default) node. Right-click and **run** the node. View the results. Sort the Terms table in the Results window by clicking the **Keep** column twice to view the terms with a Keep status Y (for Yes).

   The terms retained in the analysis are *reasonable but can be improved* by playing with start/stop lists or by developing custom synonyms list.

8. Go to the properties panel of the Text Filter (custom) node and *note* the values of **Check Spelling** to **Yes**, import the dictionary **Course.Engdict** (should be in your library), and set **Term Weight** to **Mutual Information** as shown below.



Because we have a target variable, a mutual information term weight should be used.

9. Run the Text Filter node and view the results via the interactive viewer (click the ellipsis button).

   As I have said before, this is the step where an analyst spends a significant amount of time processing the terms. Primary tasks performed at this stage include excluding irrelevant terms and creating custom synonyms. This obviously needs domain expertise. You have already seen how to create custom synonym lists in the SAS Global Forum papers. For this analysis, we are not certaing any custom synonym list. *But, results will improve if you can create a custom list using domain expertise*.

10. Use default properties for the Text Rule Builder (default) node. **Run** the node and examine the results.

    Typical predictive model fit statistics are reported in the Score Rankings Overlay, Fit Statistics, and Output windows because this node has all functionalities of other EM model nodes. From the Fit Statistics table, you will find that the model misclassification rate in training and validation data is 17.4% and 25.1% respectively. With a two-level target variable, a misclassification rate around 20% might sound reasonable. However, the big difference in training and validation data performance implies that we can possibly improve the model by modifying the rules. Other important results of the Text Rule Builder node are the rules extracted from text. The rules are nothing but key terms that were identified to be significantly associated with a particular level of target variable. These rules are listed in the Rules Obtained table. You will find terms such as **helpful**, **friendly** are being identified as rules for the target level positive, whereas **rude**, **dirty** are being associated with the target level of negative. Some of the rules use single terms, and others are conjunctions of terms and their negations (~ sign in front of a term). The ***order of rules*** listed in the table is ***very important***. The rules are applied hierarchically. That is, the second rule in the table is extracted using the documents that were not satisfied with the first rule. Similarly, the third rule is extracted using documents that were not covered by first two rules. On a scoring data set, the rules are applied in the same order.

11. Maximize the Rules Obtained window to get a sense of the terms used in predicting positive versus negative rules.

Target Value	Rule #	Rule	Precision	Recall	F1 score	Valid Precision	Valid Recall	Valid F1 score	True Positive/Total	Valid True Positive/Total
NEGATIVE	1	rude	97.96%	9.50%	17.33%	100.0%	4.72%	9.02%	96/98	13/13
NEGATIVE	2	dirty	98.59%	13.86%	24.31%	100.0%	12.60%	22.38%	53/53	21/21
NEGATIVE	3	floor	98.01%	19.50%	32.54%	97.87%	18.11%	30.56%	72/74	20/21
NEGATIVE	4	sit	98.26%	22.38%	36.45%	98.08%	20.08%	33.33%	32/32	6/6
NEGATIVE	5	filthy	98.44%	25.05%	39.94%	98.21%	21.65%	35.48%	37/37	14/14
NEGATIVE	6	disappoint	98.25%	27.72%	43.24%	98.36%	23.62%	38.10%	40/41	7/7
NEGATIVE	7	shower & ~clean & ~friendly ...	97.20%	44.75%	61.29%	92.52%	38.98%	54.85%	245/257	58/65
NEGATIVE	8	minute	96.81%	48.02%	64.20%	92.86%	40.94%	56.83%	65/70	16/16
NEGATIVE	9	poor	96.92%	49.80%	65.79%	93.04%	42.13%	57.99%	33/35	7/7
NEGATIVE	10	nasty	96.98%	50.89%	66.75%	93.10%	42.52%	58.38%	29/30	3/3
NEGATIVE	11	happen	96.91%	52.77%	68.33%	91.74%	43.70%	59.20%	35/37	9/11
NEGATIVE	12	empty	96.97%	53.86%	69.26%	91.94%	44.88%	60.32%	24/25	4/5
NEGATIVE	13	pump	96.67%	57.52%	72.13%	91.85%	48.82%	63.75%	61/71	12/15
NEGATIVE	14	bad	96.20%	60.20%	74.06%	92.31%	51.97%	66.50%	83/89	20/20
NEGATIVE	15	disgust	96.26%	61.09%	74.74%	92.31%	51.97%	66.50%	33/33	10/10
NEGATIVE	16	nasty	96.30%	61.88%	75.35%	92.41%	52.76%	67.17%	22/23	6/6
NEGATIVE	17	point	96.36%	62.97%	76.17%	92.57%	53.94%	68.16%	19/21	9/11
NEGATIVE	18	tea	96.41%	63.76%	76.76%	92.11%	55.12%	68.97%	13/13	7/8
NEGATIVE	19	three	96.45%	64.55%	77.34%	91.72%	56.69%	70.07%	19/19	5/6
NEGATIVE	20	half	96.48%	65.15%	77.78%	91.82%	57.48%	70.70%	20/22	7/7
NEGATIVE	21	right	96.52%	65.84%	78.28%	91.88%	57.87%	71.01%	13/13	5/6
NEGATIVE	22	several	96.55%	66.44%	78.71%	91.93%	58.27%	71.33%	12/12	3/3
NEGATIVE	23	mess	96.57%	66.93%	79.06%	91.93%	58.27%	71.33%	11/11	3/5
NEGATIVE	24	look	96.60%	67.52%	79.49%	92.07%	59.45%	72.25%	27/27	5/5
NEGATIVE	25	dont	96.49%	68.12%	79.86%	92.07%	59.45%	72.25%	23/24	3/3
NEGATIVE	26	fuel & ~helpful & ~great & ~g...	95.44%	72.48%	82.39%	90.06%	64.17%	74.94%	164/182	36/42
NEGATIVE	27	receipt	95.35%	73.07%	82.74%	90.06%	64.17%	74.94%	18/20	2/2
NEGATIVE	28	slow	95.37%	73.47%	83.00%	90.06%	64.17%	74.94%	14/14	2/2
NEGATIVE	29	dry	95.40%	73.86%	83.26%	89.67%	64.96%	75.34%	21/21	2/4
NEGATIVE	30	turn	95.42%	74.26%	83.52%	89.73%	65.35%	75.63%	23/24	7/8
NEGATIVE	31	report	95.44%	74.65%	83.78%	89.78%	65.75%	75.91%	9/9	1/1
POSITIVE	32	helpful	94.74%	10.76%	19.33%	95.00%	11.31%	20.21%	72/76	19/20
POSITIVE	33	great & ~bad	92.42%	29.15%	44.32%	92.00%	27.38%	42.20%	140/156	32/36
POSITIVE	34	friendly & ~floor & ~know & ~f...	92.33%	41.41%	57.17%	92.42%	36.31%	52.14%	126/137	28/29
POSITIVE	35	awesome	92.45%	43.95%	59.57%	91.55%	38.69%	54.39%	27/29	7/8
POSITIVE	36	nice	91.44%	54.26%	68.11%	90.43%	50.60%	64.89%	121/147	33/43
POSITIVE	37	excellent	91.53%	56.50%	69.87%	89.69%	51.79%	65.66%	27/30	5/6
POSITIVE	38	love	91.47%	59.34%	71.99%	88.24%	53.57%	66.67%	38/47	7/12
POSITIVE	39	good & ~customer & ~shower	90.59%	66.22%	76.51%	87.39%	61.90%	72.47%	87/112	23/35
POSITIVE	40	outstanding	90.71%	67.12%	77.15%	87.39%	61.90%	72.47%	13/13	2/2
POSITIVE	41	great	90.55%	68.76%	78.16%	87.70%	63.69%	73.79%	32/40	7/10
POSITIVE	42	kindness	90.61%	69.21%	78.47%	87.80%	64.29%	74.23%	6/6	1/1
POSITIVE	43	great	90.52%	69.96%	78.92%	86.40%	64.29%	73.72%	21/28	4/9
POSITIVE	44	best	90.53%	71.45%	79.87%	86.40%	64.29%	73.72%	24/33	5/7
POSITIVE	45	love	90.58%	71.90%	80.17%	86.61%	65.48%	74.58%	7/8	4/4
POSITIVE	46	wonderful	90.49%	72.50%	80.50%	86.61%	65.48%	74.58%	16/20	3/3

In the table above, validation dtaa metrics are reported with valid prefix in front of each metric. Metrics without valid prefix reflecst tarining data. Consider the first rule for the Positive rating. There are a total of 1,679 documents in the training data, of which 669 are rated as positive by experts in this case (you can find these numbers in the data partition results window). The first positive rule uses the term **helpful**. The numbers 72/76 in the True Positive/Total column mean that of the 1,679 documents, there were 76 documents that had the phrase **helpful** and, of those, 72 are rated positive. Using these values, we can derive precision and recall statistics. Precision measures the fraction of predicted documents that are true positives, and recall measures the fraction of actual documents that are true positives. Both these statistics use the results of the rules in the table from the first rule up to the current rule.

Close the Results window.

12. Note the setting in the **Rule Builder(custom)** node. Then right-click and **run** this node and view results.

Train	
Variables	...
Generalization Error	Very Low
Purity of Rules	Very Low
Exhaustiveness	Very High

The selection of settings above might work well when your objective is to explore the training data set without worrying about model performance on validation data. Selecting **Low** for the Generalization Error and Exhaustiveness properties will over-train the model (fits very well in the training data). Selecting **Low** for the Purity of Rules property will extract rules that handle most terms and could generate very long rules.

The Fit Statistics table shows improvement in misclassification rate in the training data set with misclassification at less than 7%. However, the misclassification rate in the validation data set is 19.6%. Such a large differential in performance is likely due to model over-training. The Rules Obtained table shows more complicated rules compared to the rules extracted with default settings. **Close** the results.

13. Go back to the **Rule Builder (default)** node. In the properties panel, click the **Change Target Values** ellipsis button to look at which comments were misclassified by the default model.

    The change target value table lists ***all wrongly classified comments***. The observations in the Change Target Values window are ordered by the model's determined "posterior probability" in descending order from 1 to 0. Therefore, the values that the model is most certain are incorrect are at the very beginning. Look at the comment that starts with "made to order burritos….." - this comment was originally rated by a human expert as EGATIVE. But if we read the comment carefully, it appears to be POSITIVE. The model has predicted this to be POSITIVE. Therefore, as part of actttttve learning, we may now modify this rating by changing the value in the last column to **NEGATIVE** from **POSITIVE**.

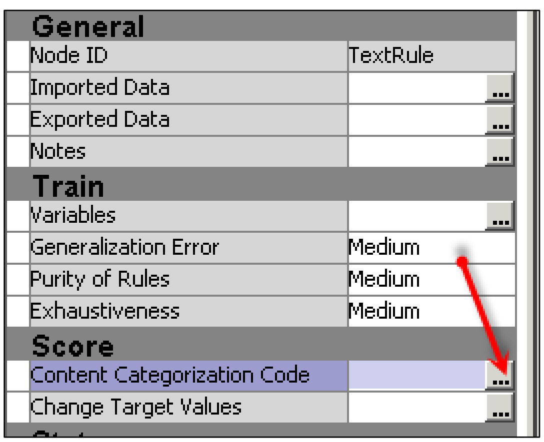
14. Click and change the Assigned Target value for this comment to **POSITIVE**.
15. **Close** the window and select **Save** when prompted.

    Any changes to the assigned target value are retained and used when the node is rerun, as long as the target variable has not been changed. When you rerun the node, your changes are applied to the data before the rule creation algorithm is run

16. Rerun the **Text Rule Builder** node and examine the results.

    The misclassification rate, as well as the number of wrong classifications in the validation data, has gone down. In practice, you will have to do this by trial and error to improve your model performance. Alternatively, you could use SAS Content Categorization Studio to improve the rules built by the Text Rules Builder node.

17. Click on the ellipsis button next to **Content Categorization Code** in the properties panel of the Text Rule Builder node.

    Examine the codes and try to make sense of the Boolean rules.

    Note the presence of positively valenced terms in the Positive rules and the negatively valenced terms in the Negative rules. Some of the rules can be made simpler and easier to understand by defining custom synonyms.

Another node that *helps us understand which words/terms are useful* to discriminate between positive and negative target values is the **Text Profile (default).** This node operates differently than Text Rule Builder. The Text Profile node enables you to profile a target variable using terms found in the documents. For each level of a target variable, the node outputs a list of terms from the collection that characterize or describe that level. The approach uses a hierarchical Bayesian model to predict which terms are the most likely terms to describe the level. In order to avoid merely selecting the most common terms, prior probabilities are used to down-weight terms that are common in more than one level of the target variable. In all cases of variable types, a corpus level profile output is also provided. This can be interpreted as the best descriptive terms for the entire collection itself. **Run** the Text Profile(default) node and view results.

18. **Run** the **Text Profile(default)** node and view results.

19. Drag the **ALL_TEST** data to diagram space and attach it to the **Score** node.
20. Run the **Score** node and examine results.

    The percentages of negative/positive are similar among train, validation, and score data. In this data set (unlike in real scoring cases), we happen to have the actual sentiment values, and those can be compared against the predicted sentiment from the Text Rule Builder model via a crosstab.

21. Attach a **SAS Code** node to the **Score** node. Go to the properties panel of the SAS Code node and click the ellipsis button next to **Code Editor**.
22. In the pop-up window, click in the Training Code white space and select **File** ⇨ **Open**. Navigate and open the file titled **Code_Chap5_Exer1** (it should be in your library). Or, you could type in the code as shown below in the Training Code window.

Macros | Macro Variables | Variables

Training Code

```
proc freq data =&EM_IMPORT_SCORE;
Tables Sentiment_Original*EM_CLASSIFICATION;
run;
```

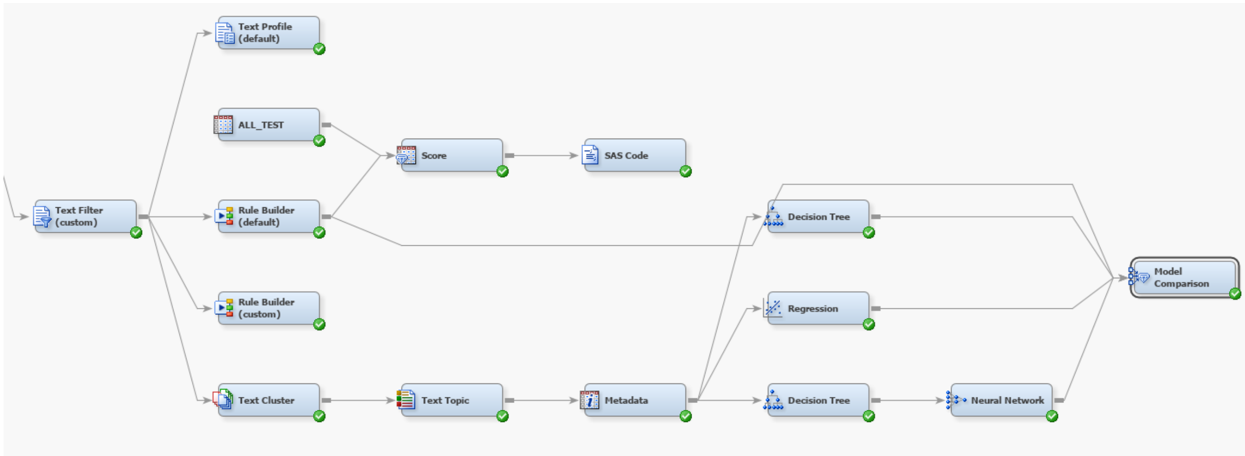
23. Close the code editor window. Save when prompted. Run the code and examine the results.

```
The FREQ Procedure

Table of Sentiment_original by EM_CLASSIFICATION

Sentiment_original
          EM_CLASSIFICATION(Prediction for Sentiment)

Frequency|
Percent  |
Row Pct  |
Col Pct  |NEGATIVE|POSITIVE|  Total
---------+--------+--------+
Negative |     93 |     48 |    141
         |  39.74 |  20.51 |  60.26
         |  65.96 |  34.04 |
         |  97.89 |  34.53 |
---------+--------+--------+
Positive |      2 |     91 |     93
         |   0.85 |  38.89 |  39.74
         |   2.15 |  97.85 |
         |   2.11 |  65.47 |
---------+--------+--------+
Total          95      139      234
            40.60    59.40   100.00
```

It seems that 93 out of 141 negative comments (66%) were correctly classified, and 91 out of 93 positive comments (98%) were also correctly classified. Overall, 186 out of 234 (78%) comments were correctly classified by the Text Rule Builder model. These are pretty good results using just the default properties of the Text Rule Builder node.

Using Text Cluster and Text Topic Nodes in Predictive Models for Document Categorization

1. Attach following nodes (**Text Cluster**, **Text Topic**, **Metadata**, **Decision Tree**, **Regression**, **Decision Tree**, **Neural Network** and **Model Comparison**) as below.



2. Make the following changes to the Text Cluster node:

Train	
Variables	...
⊟ Transform	
SVD Resolution	Low
Max SVD Dimensions	40
⊟ Cluster	
Exact or Maximum Number	Maximum
Number of Clusters	15
Cluster Algorithm	Expectation-Maximization
Descriptive Terms	15

3. Run the Text Cluster node and examine the results.

4. Make the following changes to the Text Topic node:

Train	
Variables	...
User Topics	...
⊟ Term Topics	
Number of Single-term Topics	0
⊟ Learned Topics	
Number of Multi-term Topics	15
Correlated Topics	No
⊟ Results	
Topic Viewer	...

5. Run the flow from the Text Topic node and examine the results.

6. Click the **Metadata** node to make it active. Then click the ellipsis button for **Variables Train** in the properties panel of the Metadata node.

7. Sort the table by **Name**. Make the changes to the variable roles as shown below. Most of the changes in the New Role columns have been highlighted below. Essentially, you are changing **Comment** to **Rejected**, **TextCluster_cluster_** to **Input** (from **Segment**), and all **TextTopic_1** through **TextTopic_15** to **Input** (from **Segment**).

Variables - Meta

(none) | not | Equal to

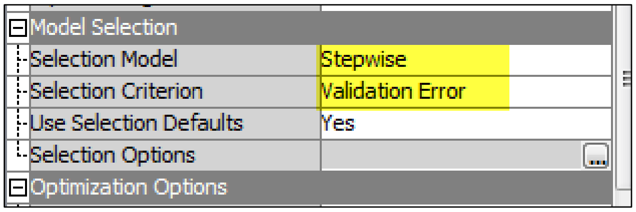
Columns: Label | Mining | Basic

Name	Hidden	Hide	Role	New Role	Level	New Level	New Order	New Report
dataobs	N	Default	ID	Default	Interval	Default	Default	Default
DOCUMENT	N	Default	ID	Default	Nominal	Default	Default	Default
Comment	N	Default	Text	Rejected	Nominal	Default	Default	Default
ID	N	Default	ID	Default	Nominal	Default	Default	Default
Sentiment	N	Default	Target	Default	Nominal	Default	Default	Default
TextCluster_cluster_	N	Default	Segment	Input	Nominal	Default	Default	Default
TextCluster_prob1	N	Default	Rejected	Default	Interval	Default	Default	Default
TextCluster_prob10	N	Default	Rejected	Default	Interval	Default	Default	Default
TextCluster_prob2	N	Default	Rejected	Default	Interval	Default	Default	Default
TextCluster_prob3	N	Default	Rejected	Default	Interval	Default	Default	Default
TextCluster_prob4	N	Default	Rejected	Default	Interval	Default	Default	Default
TextCluster_prob5	N	Default	Rejected	Default	Interval	Default	Default	Default
TextCluster_prob6	N	Default	Rejected	Default	Interval	Default	Default	Default
TextCluster_prob7	N	Default	Rejected	Default	Interval	Default	Default	Default
TextCluster_prob8	N	Default	Rejected	Default	Interval	Default	Default	Default
TextCluster_prob9	N	Default	Rejected	Default	Interval	Default	Default	Default
TextCluster_SVD1	N	Default	Input	Default	Interval	Default	Default	Default
TextCluster_SVD10	N	Default	Input	Default	Interval	Default	Default	Default
TextCluster_SVD11	N	Default	Input	Default	Interval	Default	Default	Default
TextCluster_SVD12	N	Default	Input	Default	Interval	Default	Default	Default
TextCluster_SVD13	N	Default	Input	Default	Interval	Default	Default	Default
TextCluster_SVD14	N	Default	Input	Default	Interval	Default	Default	Default
TextCluster_SVD2	N	Default	Input	Default	Interval	Default	Default	Default
TextCluster_SVD3	N	Default	Input	Default	Interval	Default	Default	Default
TextCluster_SVD4	N	Default	Input	Default	Interval	Default	Default	Default
TextCluster_SVD5	N	Default	Input	Default	Interval	Default	Default	Default
TextCluster_SVD6	N	Default	Input	Default	Interval	Default	Default	Default
TextCluster_SVD7	N	Default	Input	Default	Interval	Default	Default	Default
TextCluster_SVD8	N	Default	Input	Default	Interval	Default	Default	Default
TextCluster_SVD9	N	Default	Input	Default	Interval	Default	Default	Default
TextTopic_1	N	Default	Segment	Input	Binary	Default	Default	Default
TextTopic_10	N	Default	Segment	Input	Binary	Default	Default	Default
TextTopic_11	N	Default	Segment	Input	Binary	Default	Default	Default
TextTopic_12	N	Default	Segment	Input	Binary	Default	Default	Default
TextTopic_13	N	Default	Segment	Input	Binary	Default	Default	Default
TextTopic_14	N	Default	Segment	Input	Binary	Default	Default	Default
TextTopic_15	N	Default	Segment	Input	Binary	Default	Default	Default
TextTopic_2	N	Default	Segment	Input	Binary	Default	Default	Default
TextTopic_3	N	Default	Segment	Input	Binary	Default	Default	Default
TextTopic_4	N	Default	Segment	Input	Binary	Default	Default	Default
TextTopic_5	N	Default	Segment	Input	Binary	Default	Default	Default
TextTopic_6	N	Default	Segment	Input	Binary	Default	Default	Default
TextTopic_7	N	Default	Segment	Input	Binary	Default	Default	Default
TextTopic_8	N	Default	Segment	Input	Binary	Default	Default	Default
TextTopic_9	N	Default	Segment	Input	Binary	Default	Default	Default
TextTopic_raw1	N	Default	Input	Default	Interval	Default	Default	Default
TextTopic_raw10	N	Default	Input	Default	Interval	Default	Default	Default
TextTopic_raw11	N	Default	Input	Default	Interval	Default	Default	Default

8. Change **Assessment Measure** (under Subtree) for the *stand-alone decision tree* to **Average Square Error** as shown below.

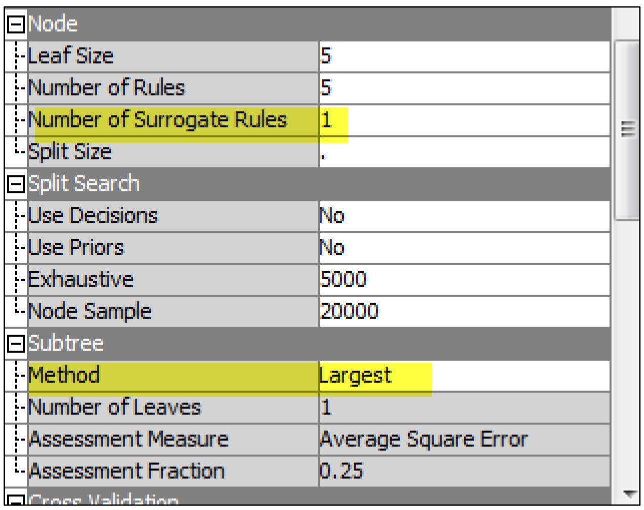
Subtree	
Method	Assessment
Number of Leaves	1
Assessment Measure	Average Square Error
Assessment Fraction	0.25

9. For the Regression node, change the selection model to **Stepwise** and the selection criterion to **Validation Error** as shown below.



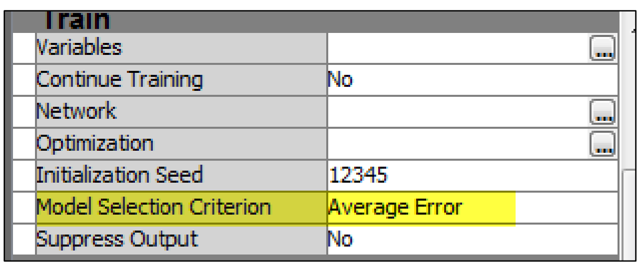
Model Selection	
Selection Model	Stepwise
Selection Criterion	Validation Error
Use Selection Defaults	Yes
Selection Options	...
Optimization Options	

10. The *Decision Tree connected to the Neural Network* will be used as a *variable selection tree*. Change the properties of this tree so that the number of surrogate rules is **1** and the method is **Largest** as shown below.



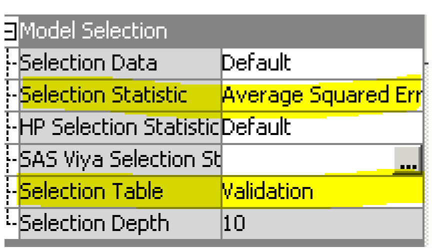
Node	
Leaf Size	5
Number of Rules	5
Number of Surrogate Rules	1
Split Size	.
Split Search	
Use Decisions	No
Use Priors	No
Exhaustive	5000
Node Sample	20000
Subtree	
Method	Largest
Number of Leaves	1
Assessment Measure	Average Square Error
Assessment Fraction	0.25
Cross Validation	

11. In the Neural Network node, change **Model Selection Criterion** to **Average Error** as shown below.



Train	
Variables	...
Continue Training	No
Network	...
Optimization	...
Initialization Seed	12345
Model Selection Criterion	Average Error
Suppress Output	No

12. In the properties panel for the Model Comparison node, change **Selection Statistic** to **Average Squared Error** and **Selection Table** to **Validation** as shown below.



Model Selection	
Selection Data	Default
Selection Statistic	Average Squared Err
HP Selection Statistic	Default
SAS Viya Selection St	...
Selection Table	Validation
Selection Depth	10

13. Right-click and run the flow from the Model Comparison node. Examine the results.

It seems that in this data the Text Rule Builder model has outperformed all of the other models using the chosen criteria (ASE) for comparing models. However, it's a different story if we use say validation ROC or Validation KS statistic.

Using Numeric and Textual Data for Predictive Modeling

Case Study: Improving Predictive Model Using Textual Data

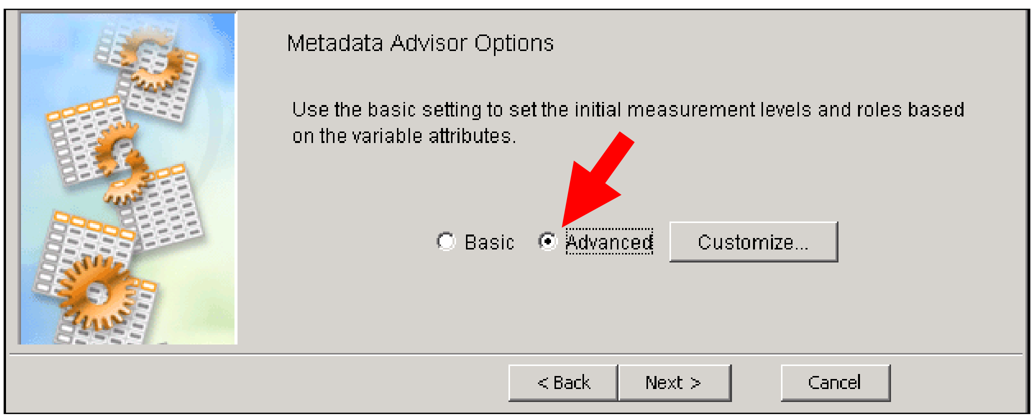
Case Description

The data used in the case study are created based on a real data set of a client company (Fuel Stop Company with over 300 gas stations in the United States). Some of the text comments, variable names, and descriptions have been disguised to protect the identity of the client company and the actual nature of the project. However, you can make out the general nature of the variables by their names. The case involves customers calling the fuel company's call center for many different reasons. Customers' comments via phone were captured by call center reps and typed in a form. These comments were later merged with numeric variables from the fuel company's database about these customers (by matching them via the company's loyalty card number).

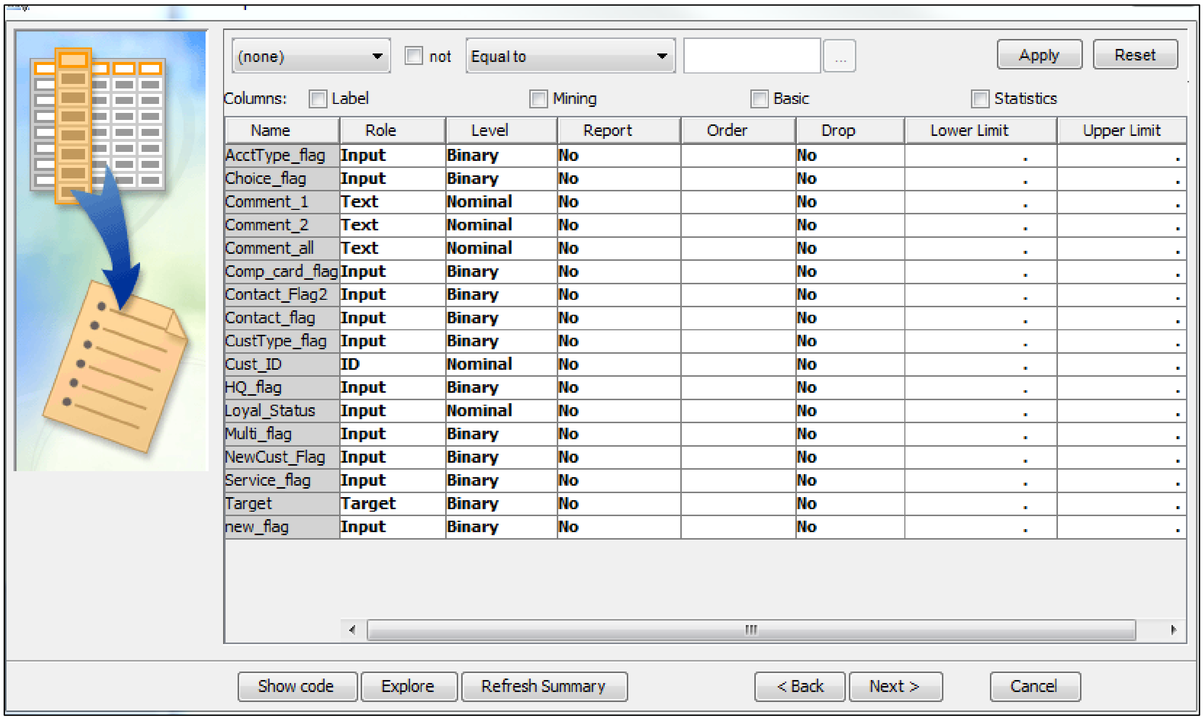
The merged data set (**GAS_TEXT_NUMERIC_DATA**) is being used in this case study. The purpose of this case study is to demonstrate how the use of textual data in conjunction with numeric data in a predictive model improves the performance of the predictive model.

Note: In the steps below, we open an already created diagram ***gas_text_numeric_predmdel(nodata)*** and then create a library (name it Course) and add the data source to this digram.

1. Create a new project or start with an existing project.
2. Right-click diagrams in the project panel and select Import Diagram from XML. Select he diagram ***gas_text_numeric_predmdel(nodata)***
3. Create a library (name it as Course)to point to a folder where the data are located. Add the data source, **gas_text_numeric_data**, to the project (via your library).
4. In Step 4, ensure that you select **Advanced** under Metadata Advisor Options as shown below.



5. The variable roles and levels are shown below.

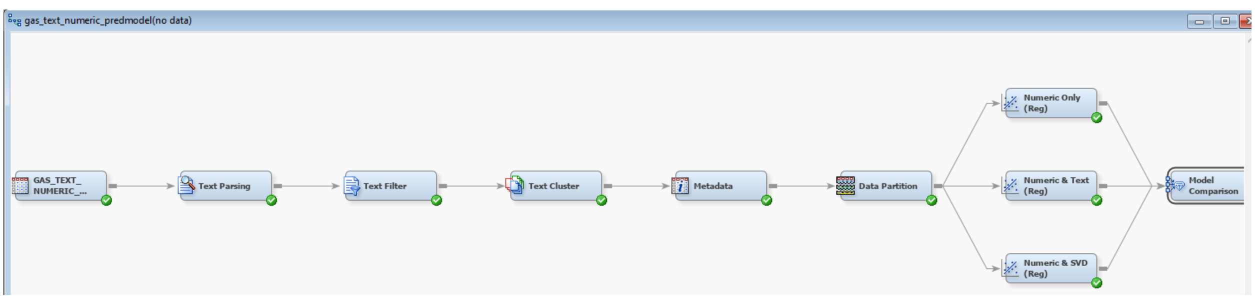


Name	Role	Level	Report	Order	Drop	Lower Limit	Upper Limit
AcctType_flag	Input	Binary	No		No	.	.
Choice_flag	Input	Binary	No		No	.	.
Comment_1	Text	Nominal	No		No	.	.
Comment_2	Text	Nominal	No		No	.	.
Comment_all	Text	Nominal	No		No	.	.
Comp_card_flag	Input	Binary	No		No	.	.
Contact_Flag2	Input	Binary	No		No	.	.
Contact_flag	Input	Binary	No		No	.	.
CustType_flag	Input	Binary	No		No	.	.
Cust_ID	ID	Nominal	No		No	.	.
HQ_flag	Input	Binary	No		No	.	.
Loyal_Status	Input	Nominal	No		No	.	.
Multi_flag	Input	Binary	No		No	.	.
NewCust_Flag	Input	Binary	No		No	.	.
Service_flag	Input	Binary	No		No	.	.
Target	Target	Binary	No		No	.	.
new_flag	Input	Binary	No		No	.	.

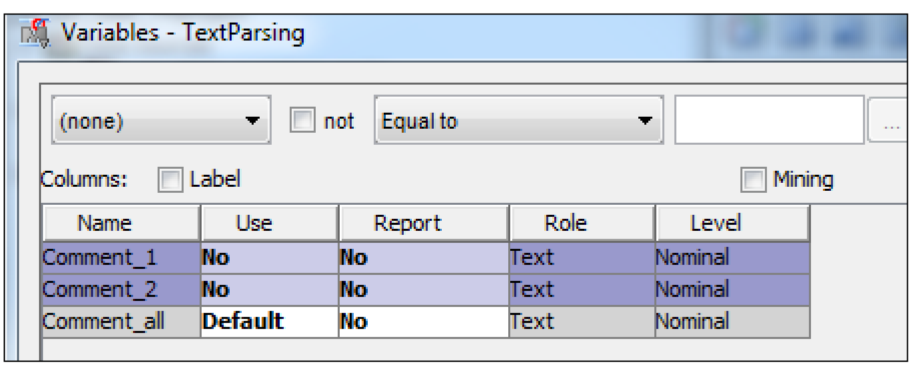
6. Click through and finish the next data creation steps by accepting the default options.

7. Drag the **gas_text_numeric_data** data to the diagram space.

8. **Add** the data source to Text Parsing node.



9. Right-click the **Text Parsing** node and select **Edit Variables**. Note that the Use role for **Comment_1** and **Comment_2** has been changed to **No**.



Name	Use	Report	Role	Level
Comment_1	No	No	Text	Nominal
Comment_2	No	No	Text	Nominal
Comment_all	Default	No	Text	Nominal

In this case study, you are using all of the comments together to create text clusters. It is, however, possible to create clusters separately for **Comment_1** and **Comment_2** and perhaps you should explore that on your own as a self-study.

10. Right-click the **Text Cluster** node and examine the results.

    You will find that there are many small clusters with few observations when the Text Cluster node is run with its default settings. This is not surprising given the small data set.

11. The following highlighted changes have been made in the properties panel of the Text Cluster node. Given small data size, for demonstration, we will ask SAS Text Miner to create a maximum of 20 SVD dimensions and exactly 5 clusters and describe those clusters using 10 terms

Train	
Variables	...
Transform	
SVD Resolution	Low
Max SVD Dimensions	20
Cluster	
Exact or Maximum Number	Exact
Number of Clusters	5
Cluster Algorithm	Expectation-Maximization
Descriptive Terms	10

12. Right-click the **Text Cluster** node and examine the results.

    You should explore the cluster solution to get a feel for what these clusters might represent. You can use a Segment Profile node to profile these clusters using the numeric variables in the data.

13. In the **Metadata** node, click the ellipsis button next to **Train** in the Variables section of the properties panel of the metadata. Then note the following changes as shown below.

Name	Hidden	Hide	Role	New Role	Level	New Level	New Order	New Report
AcctType_flag	N	Default	Input	Default	Binary	Default	Default	Default
Choice_flag	N	Default	Input	Default	Binary	Default	Default	Default
Comment_1	N	Default	Text	Default	Nominal	Default	Default	Default
Comment_2	N	Default	Text	Default	Nominal	Default	Default	Default
Comment_all	N	Default	Text	Default	Nominal	Default	Default	Default
Comp_card_flag	N	Default	Input	Default	Binary	Default	Default	Default
Contact_Flag2	N	Default	Input	Default	Binary	Default	Default	Default
Contact_flag	N	Default	Input	Default	Binary	Default	Default	Default
CustType_flag	N	Default	Input	Default	Binary	Default	Default	Default
Cust_ID	N	Default	ID	Default	Nominal	Default	Default	Default
HQ_flag	N	Default	Input	Default	Binary	Default	Default	Default
Loyal_Status	N	Default	Input	Default	Nominal	Default	Default	Default
Multi_flag	N	Default	Input	Default	Binary	Default	Default	Default
NewCust_Flag	N	Default	Input	Default	Binary	Default	Default	Default
Service_flag	N	Default	Input	Default	Binary	Default	Default	Default
Target	N	Default	Target	Default	Binary	Default	Default	Default
TextCluster_SVD1	N	Default	Input	Default	Interval	Default	Default	Default
TextCluster_SVD2	N	Default	Input	Default	Interval	Default	Default	Default
TextCluster_SVD3	N	Default	Input	Default	Interval	Default	Default	Default
TextCluster_SVD4	N	Default	Input	Default	Interval	Default	Default	Default
TextCluster_SVD5	N	Default	Input	Default	Interval	Default	Default	Default
TextCluster_SVD6	N	Default	Input	Default	Interval	Default	Default	Default
TextCluster_SVD7	N	Default	Input	Default	Interval	Default	Default	Default
TextCluster_SVD8	N	Default	Input	Default	Interval	Default	Default	Default
TextCluster_SVD9	N	Default	Input	Default	Interval	Default	Default	Default
TextCluster_cluster_	N	Default	Segment	Input	Nominal	Default	Default	Default
TextCluster_prob1	N	Default	Rejected	Default	Interval	Default	Default	Default
TextCluster_prob2	N	Default	Rejected	Default	Interval	Default	Default	Default
TextCluster_prob3	N	Default	Rejected	Default	Interval	Default	Default	Default
TextCluster_prob4	N	Default	Rejected	Default	Interval	Default	Default	Default
TextCluster_prob5	N	Default	Rejected	Default	Interval	Default	Default	Default
document	N	Default	ID	Default	Nominal	Default	Default	Default
new_flag	N	Default	Input	Default	Binary	Default	Default	Default

14. Add a Data Partition node from the Sample tab to the Metadata node.

15. The following changes were made in the properties panel of the Data Partition node under Data Set Allocations: Training is set to **80**, Validation to **20** and Test to **0**.

Property	Value
General	
Node ID	Part
Imported Data	...
Exported Data	...
Notes	...
Train	
Variables	...
Output Type	Data
Partitioning Method	Default
Random Seed	12345
Data Set Allocations	
Training	80.0
Validation	20.0
Test	0.0
Report	
Interval Targets	Yes
Class Targets	Yes

16. From the Model tab, a **Regression** node has been connected the **Data Partition** node. This node has been renamed as **Numeric Only (Reg)**.
17. Right-click the **Numeric Only (Reg)** node and select **Edit variables**.
18. Note the change in the Use role of all cluster variables to **No**. Click **OK**.

(none) not Equal to ...

Columns: Label Minir

Name	Use	Report	Role	Level
AcctType_flag	Default	No	Input	Binary
Choice_flag	Default	No	Input	Binary
Comp_card_flag	Default	No	Input	Binary
Contact_Flag2	Default	No	Input	Binary
Contact_flag	Default	No	Input	Binary
CustType_flag	Default	No	Input	Binary
HQ_flag	Default	No	Input	Binary
Loyal_Status	Default	No	Input	Nominal
Multi_flag	Default	No	Input	Binary
NewCust_Flag	Default	No	Input	Binary
Service_flag	Default	No	Input	Binary
Target	Yes	No	Target	Binary
TextCluster_SVD1	No	No	Input	Interval
TextCluster_SVD2	No	No	Input	Interval
TextCluster_SVD3	No	No	Input	Interval
TextCluster_SVD4	No	No	Input	Interval
TextCluster_SVD5	No	No	Input	Interval
TextCluster_SVD6	No	No	Input	Interval
TextCluster_SVD7	No	No	Input	Interval
TextCluster_SVD8	No	No	Input	Interval
TextCluster_cluster_	No	No	Input	Nominal
TextCluster_prob1	No	No	Rejected	Interval
TextCluster_prob2	No	No	Rejected	Interval
TextCluster_prob3	No	No	Rejected	Interval
TextCluster_prob4	No	No	Rejected	Interval
TextCluster_prob5	No	No	Rejected	Interval
new_flag	Default	No	Input	Binary

19. In the properties panel of the Numeric Only (Reg) node, the following changes have been made: the selection model is set to **Stepwise** and the selection criterion is set to **Validation Error**.

Train	
Variables	...
Equation	
Main Effects	Yes
Two-Factor Interactions	No
Polynomial Terms	No
Polynomial Degree	2
User Terms	No
Term Editor	...
Class Targets	
Regression Type	Logistic Regression
Link Function	Logit
Model Options	
Suppress Intercept	No
Input Coding	Deviation
Model Selection	
Selection Model	Stepwise
Selection Criterion	Validation Error
Use Selection Defaults	Yes
Selection Options	...

20. In the diagram space, the Numeric Only (Reg) node has been copied and pasted. The name for the pasted node has been changed to **Numeric & Text (Reg)** and connected with the data partition node.
21. Right-click the **Numeric & Text (Reg)** node and select **Edit variables**.
22. Note the change to the Use role of the cluster membership variable from **No** to **Default** as shown below. Then click **OK**.

Columns: ☐ Label

Name	Use	Report	Role	Level
AcctType_flag	**Default**	**No**	Input	Binary
Choice_flag	**Default**	**No**	Input	Binary
Comp_card_flag	**Default**	**No**	Input	Binary
Contact_Flag2	**Default**	**No**	Input	Binary
Contact_flag	**Default**	**No**	Input	Binary
CustType_flag	**Default**	**No**	Input	Binary
HQ_flag	**Default**	**No**	Input	Binary
Loyal_Status	**Default**	**No**	Input	Nominal
Multi_flag	**Default**	**No**	Input	Binary
NewCust_Flag	**Default**	**No**	Input	Binary
Service_flag	**Default**	**No**	Input	Binary
Target	**Yes**	**No**	Target	Binary
TextCluster_SVD1	**No**	**No**	Input	Interval
TextCluster_SVD2	**No**	**No**	Input	Interval
TextCluster_SVD3	**No**	**No**	Input	Interval
TextCluster_SVD4	**No**	**No**	Input	Interval
TextCluster_SVD5	**No**	**No**	Input	Interval
TextCluster_SVD6	**No**	**No**	Input	Interval
TextCluster_SVD7	**No**	**No**	Input	Interval
TextCluster_SVD8	**No**	**No**	Input	Interval
TextCluster_cluster_	**Default**	**No**	Input	Nominal
TextCluster_prob1	**No**	**No**	Rejected	Interval
TextCluster_prob2	**No**	**No**	Rejected	Interval
TextCluster_prob3	**No**	**No**	Rejected	Interval
TextCluster_prob4	**No**	**No**	Rejected	Interval
TextCluster_prob5	**No**	**No**	Rejected	Interval
new_flag	**Default**	**No**	Input	Binary

22. In the diagram space, the Numeric Only (Reg) node has been copied and pasted. The name of the pasted node has been changed to **Numeric & SVD (Reg)** and connected to the **Data Partition** node.
23. Right-click the **Numeric & SVD (Reg)** node and select **Edit variables**.

24. Note the change in the **Use** role of the SVD variables from **No** to **Default**. Click **OK**.

Columns: ☐ Label ☐ Mining

Name	Use	Report	Role	Level
AcctType_flag	Default	No	Input	Binary
Choice_flag	Default	No	Input	Binary
Comp_card_flag	Default	No	Input	Binary
Contact_Flag2	Default	No	Input	Binary
Contact_flag	Default	No	Input	Binary
CustType_flag	Default	No	Input	Binary
HQ_flag	Default	No	Input	Binary
Loyal_Status	Default	No	Input	Nominal
Multi_flag	Default	No	Input	Binary
NewCust_Flag	Default	No	Input	Binary
Service_flag	Default	No	Input	Binary
Target	Yes	No	Target	Binary
TextCluster_SVD1	Default	No	Input	Interval
TextCluster_SVD2	Default	No	Input	Interval
TextCluster_SVD3	Default	No	Input	Interval
TextCluster_SVD4	Default	No	Input	Interval
TextCluster_SVD5	Default	No	Input	Interval
TextCluster_SVD6	Default	No	Input	Interval
TextCluster_SVD7	Default	No	Input	Interval
TextCluster_SVD8	Default	No	Input	Interval
TextCluster_SVD9	Default	No	Input	Interval
TextCluster_cluster_	No	No	Input	Nominal
TextCluster_prob1	No	No	Rejected	Interval
TextCluster_prob2	No	No	Rejected	Interval
TextCluster_prob3	No	No	Rejected	Interval
TextCluster_prob4	No	No	Rejected	Interval
TextCluster_prob5	No	No	Rejected	Interval
new_flag	Default	No	Input	Binary

25. A Model Comparison node (from the Assess tab) has been connected with all Regression nodes.

26. Note the changes in the properties of the Model Selection node.

Model Selection	
Selection Data	Default
Selection Statistic	Average Squared Error
HP Selection Statistic	Default
SAS Viya Selection Statis	
Selection Table	Validation
Selection Depth	10

27. Right-click the **Model Comparison** node and examine the results.

Notice that for the validation data, the **Numeric & SVD (Reg)** model has clearly outperformed the **Numeric & Text (Reg) model,** which has outperformed the **Numeric Only (Reg)** model. Thus, additions of SVDS or text clusters (or both) have improved the predictive ability of the model over a model that has only numeric variables.

Self-Study:

Explore different panels in the Results window of the Model Comparison node and regression results on your own. Then do following:

- Attach a Text Topic node to the Text Cluster node. Use the **Text Topics** as additional input variables along with text cluster input variables in the predictive models. (Make sure that you change the ***Text_Topic variables roles to Input via a Metadata node*** as I demonstrated in the last example. Explore whether those changes improve the model.
- Try other predictive models such as decision tree and neural net on this data and see if that can improve model prediction.