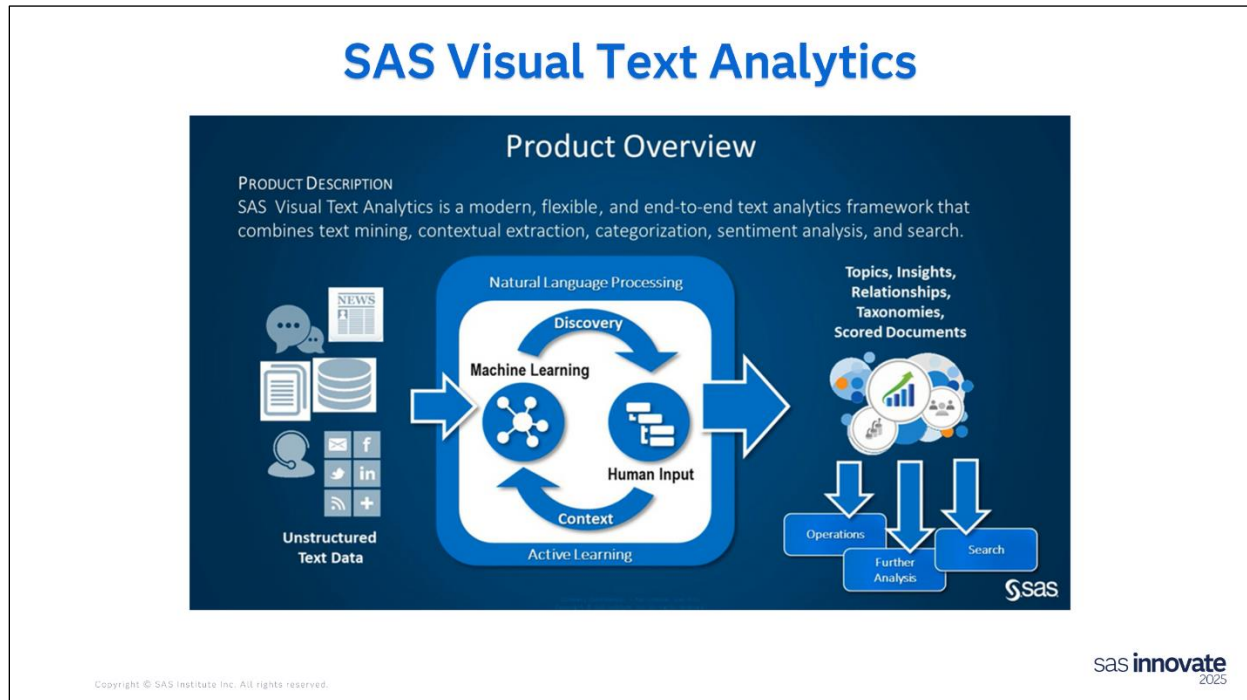


Lesson 1 Hands-on Workshop: SAS[®] Visual Text Analytics in SAS[®] Viya[®]

Jeffrey Thompson, PhD
Sr. Analytical Training Consultant

1.1 Introduction to SAS Visual Text Analytics



SAS Visual Text Analytics is a high-end software product that simplifies the use of text analytics for many types of users, from business analysts to document librarians. By choosing intelligent parameter settings, or by using machine-learning tools to “learn” an appropriate setting, Visual Text Analytics frees the user to concentrate on the immediate task. The downside to making the software powerful and easy to use is that users who try to explore specific algorithmic details for educational purposes are limited in the experiments that can be designed to help understand algorithmic choices.

SAS Visual Text Analytics is a web-based, text analytics application that enables you to identify key terms and concepts in your document collections, build concept and topic models, and use linguistic rules to categorize documents.

SAS Visual Text Analytics can be accessed via a point-and-click interface included in SAS Viya called Model Studio. Model Studio will be used to access Visual Text Analytics in the following demonstrations. Visual Text Analytics procedures and CAS actions can also be used for those more comfortable working with code. SAS Studio can be used as a code-editor interface in such cases.

SAS Visual Text Analytics helps the text analyst face the big, unstructured text data challenges effectively in a timely manner by providing powerful tools in the fields of text data exploration and visualization, information retrieval, and content categorization.

SAS Visual Text Analytics: Capabilities and Benefits

- **Natural language processing:** Enhances parsing to add language features and expand the document collection term table; supports topic derivation
- **Automated feature extraction with machine-generated topics:** Discovers themes and shows related terms and documents for each theme
- **Native linguistic support for multiple languages:** Supports the global nature of a business (more than 30 world languages)
- **Sentiment analysis:** Supports business decisions by revealing trending perspectives
- **Support for both machine learning and rules-based approaches within a single project:** Enables scoring of new documents to reveal emerging trends and identify dominant themes

Copyright © SAS Institute Inc. All rights reserved.

sas **innovate**
2025

SAS Visual Text Analytics: Capabilities and Benefits

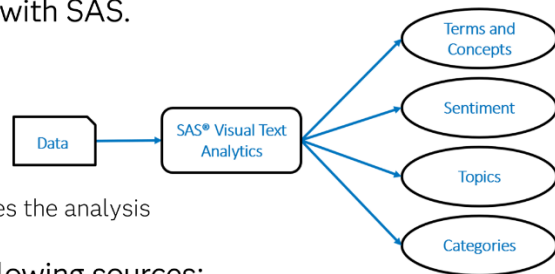
- **Contextual extraction:** Enables non-ambiguous coding of subject matter expertise to extract specific information from within documents
- **Flexible deployment:** Maximizes the data's value and accelerates the data to the decision timeline
- **Facilitation of collaboration in a multi-user environment:** Fuels collaboration and information sharing in an open analytics ecosystem

Copyright © SAS Institute Inc. All rights reserved.

sas **innovate**
2025

SAS Visual Text Analytics: Big Picture

- SAS Visual Text Analytics is integrated with SAS.
 - accessible via Model Studio in SAS Viya
 - input or score data
- You can use the product to manage multiple projects.
 - An interactive point-and-click browser GUI guides the analysis of large or complex text data.
- You can easily include data from the following sources:
 - SAS table or CAS table
 - Documents converted using SAS Data Explorer



Copyright © SAS Institute Inc. All rights reserved.

sas **innovate**
2025

Manual text analysis efforts suffer from inadequacy due to human subjectivity and inconsistency, as well as the time that is required to read each document and classify it. SAS Visual Text Analytics eliminates the need to manually review documents, develop a training corpus, and manually develop taxonomies. After data are registered to the software, natural language processing (NLP) is automatically performed. This includes tokenization, term frequency counts, stemming, and part-of-speech tagging. Combining statistical machine learning with an extensive array of linguistic operators and prebuilt concept definitions, the text analyst is empowered to customize the automatically discovered results within a single, visual, guided application.

From a practical standpoint, SAS Visual Text Analytics is a single application that brings together the techniques that are used in text mining, categorization, contextual extraction, sentiment analysis, and topic derivation. This enables analysts to apply the appropriate analysis to meet their specific use cases without needing to switch applications or move data around. Using SAS Visual Text Analytics is more productive when it is no longer economical to manually review and classify your volume of documents (typically, greater than 500 documents) or when errors associated with manual tagging result in inconsistent, untrustworthy, or misinformed business understanding.



Creating a SAS Visual Text Analytics Project with No Predefined Concepts

This demonstration introduces SAS Visual Text Analytics features that enable users to automatically extract topics and develop tools that automatically classify categories of interest without using any predefined concepts.

This demonstration has four objectives:

- exploring and preparing a document collection
- using the default functionality to create a SAS Visual Text Analytics project with the demonstration data
- exploring the automatically generated topics and the associated documents, and promoting the most intriguing topics to categories
- exploring the rules that are generated by SAS Visual Text Analytics for identifying and categorizing the documents that belong to the relevant categories

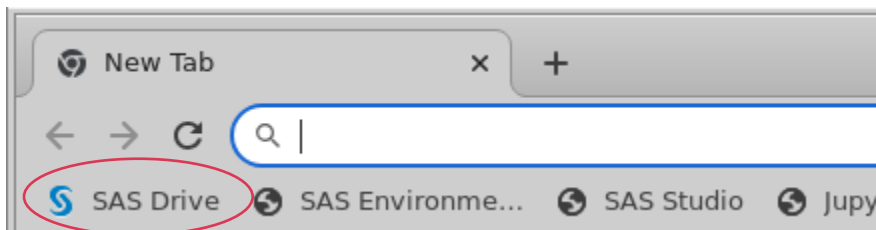
Using SAS tools such as SAS Visual Analytics to explore text data prior to building text models is highly recommended. For brevity, we skip this initial exploration in the current demonstration.

The data set for this demonstration is **drug_reports**. The data set contains 1,414 patient comments about drug side effects. The patients are on prescription drug medications to treat depression and anxiety. The end goal of the analysis is to use the Topics node to search for prevailing themes within the patient feedback documents and promote any topics related to positive patient recovery as categories.


Creating a SAS Visual Text Analytics Project (Step 1 of 3)

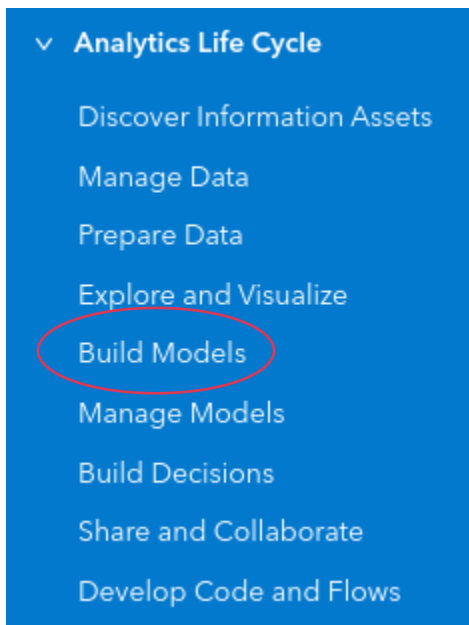
Follow the steps below to create the Visual Text Analytics project (step 1).

1. Open Google Chrome from the virtual computer desktop.
2. Using the shortcuts at the top of the browser window, select **SAS Drive**.



3. Sign in using these credentials:
User ID: **student**
Password: **Metadata0**
4. Click **Yes** to opt in to assumable groups.

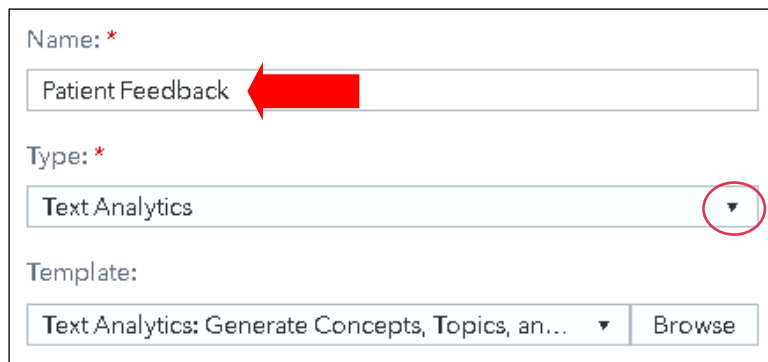
5. In the top left of SAS Drive, click the **applications menu** button  and select **Build Models**. This action invokes Model Studio. Note that the Model Studio interface also has an applications menu button.



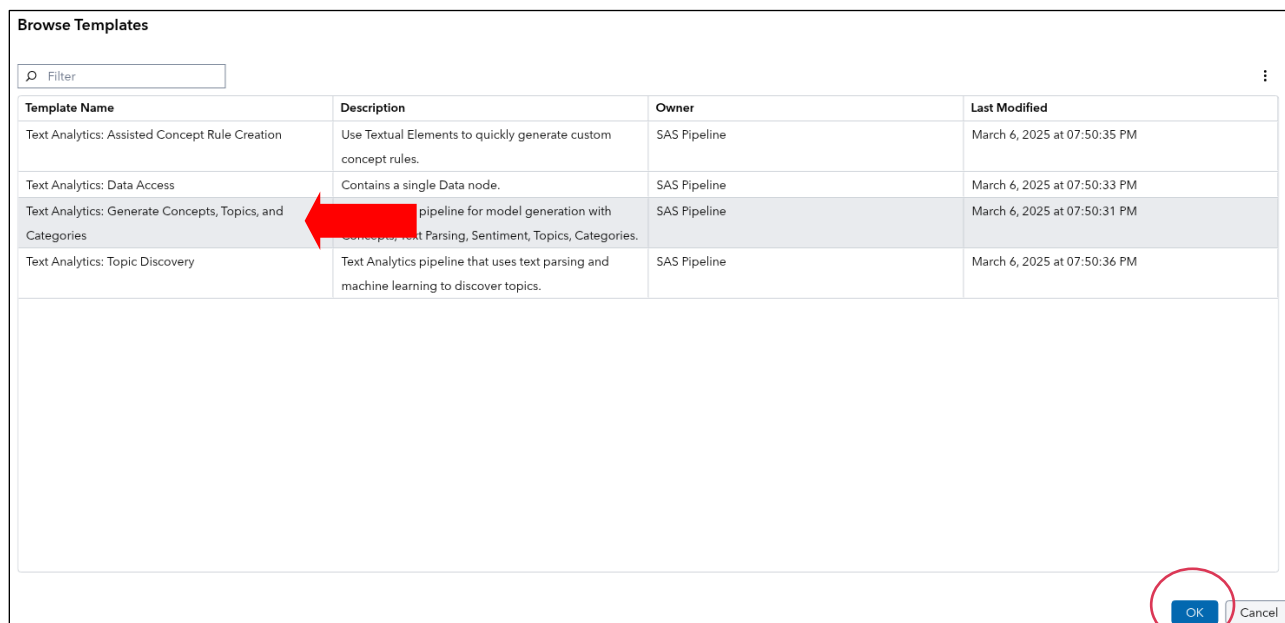
6. In Model Studio, click **New Project** in the upper right corner.



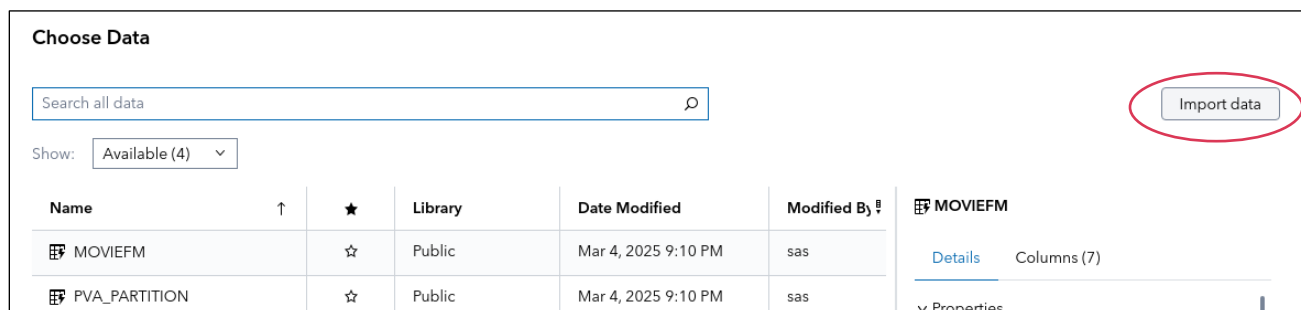
7. Enter **Patient Feedback** for **Name** and select **Text Analytics** from the drop-down menu under **Type**.



8. Select **Browse** under **Template**. In the Browse Template window, select the Template Named **Text Analytics: Generate Concepts, Topics, and Categories**. Click **OK**.



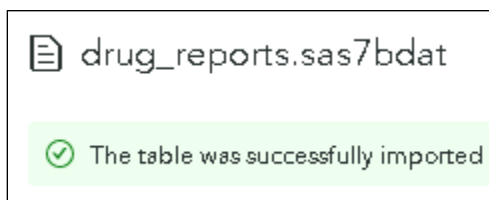
9. Select **Browse** under **Data**. The Choose Data window appears, showing available data, that is pre-loaded into CAS memory. We must import our data set. Click **Import data**.



10. Select **Local Files**.

11. Navigate to **workshop > SIWTAS_Text**. Select **drug_reports.sas7bdat** and click **Open**.

12. In the Import Data window, click **Import Item**. The green check mark indicates that the data were successfully loaded into memory.



13. Click **Add**. Select **DRUG_REPORTS** from the Available list and then click **OK**.

The New Project window should appear as follows. Click **Save**.

New Project

Name: *
Patient Feedback

Type: *
Text Analytics

Template:
Text Analytics: Generate Concepts, Topics, an... Browse

Data: *
Public.DRUG_REPORTS Browse

Description:

Save Cancel

14. Once the project is saved, it appears on the Data tab. The Data tab displays the variables of the input data and enables you to assign certain metadata roles. Model Studio shows a warning at the top of the page indicating that a Text variable must be assigned.

Model Studio - Build Models

Patient Feedback

Data Pipelines

Data sources

Project data table

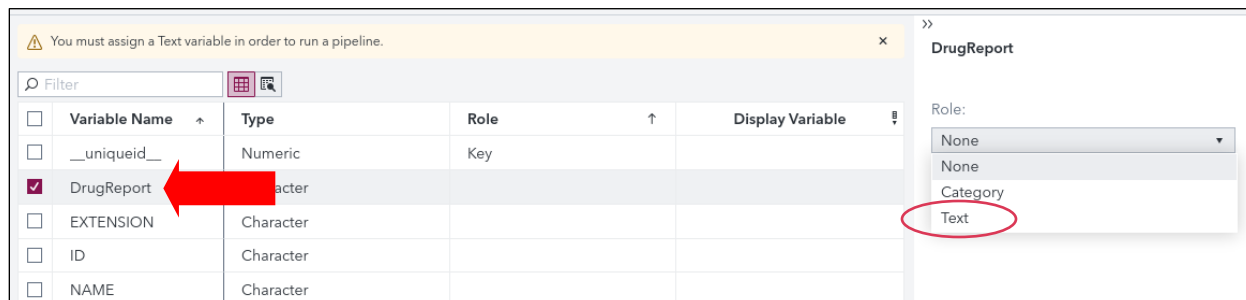
DRUG_REPORTS

Filter

<input type="checkbox"/>	Variable Name	Type	Role	Display Variable
<input type="checkbox"/>	__uniqueid__	Numeric	Key	
<input type="checkbox"/>	DrugReport	Character		
<input type="checkbox"/>	EXTENSION	Character		
<input type="checkbox"/>	ID	Character		
<input type="checkbox"/>	NAME	Character		

You must assign a Text variable in order to run a pipeline.

15. Select **DrugReport** by clicking the check box next to the variable name. Using the pane on the right, assign the role **Text** for this variable. Keep all other properties in the properties pane at their default settings.

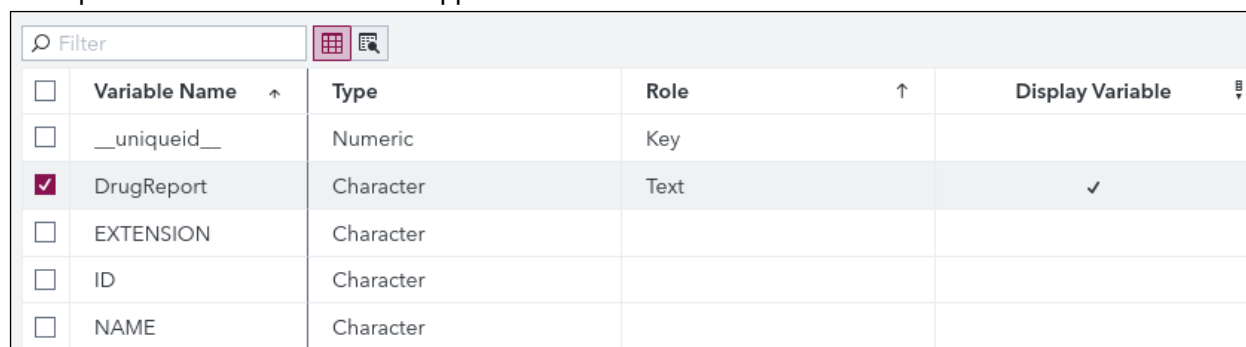
 The screenshot shows the SAS Visual Text Analytics interface. On the left, a table lists variables: Variable Name, Type, Role, and Display Variable. The variable 'DrugReport' is selected with a checkmark. A red arrow points to the 'DrugReport' row. On the right, the 'DrugReport' properties pane is open, showing the 'Role' dropdown menu. The 'Text' role is selected and circled in red.

Variable Name	Type	Role	Display Variable
<input type="checkbox"/> __uniqueid__	Numeric	Key	
<input checked="" type="checkbox"/> DrugReport	Character		
<input type="checkbox"/> EXTENSION	Character		
<input type="checkbox"/> ID	Character		
<input type="checkbox"/> NAME	Character		

Role:
 None
 None
 Category
 Text

Note: The other variable role is for Category. No Category variables exist in the **Drug_Reports** data, so this role assignment is not needed for this project.

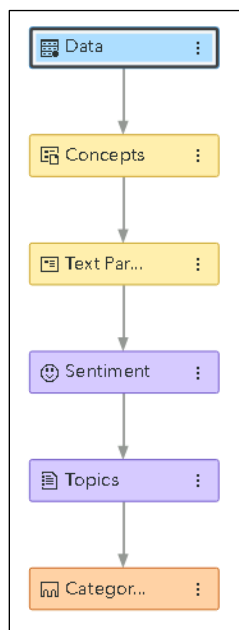
The updated variables list should appear as follows.

 The screenshot shows the updated variable list. The 'DrugReport' variable is now assigned the role 'Text' and has a checkmark in the 'Display Variable' column.

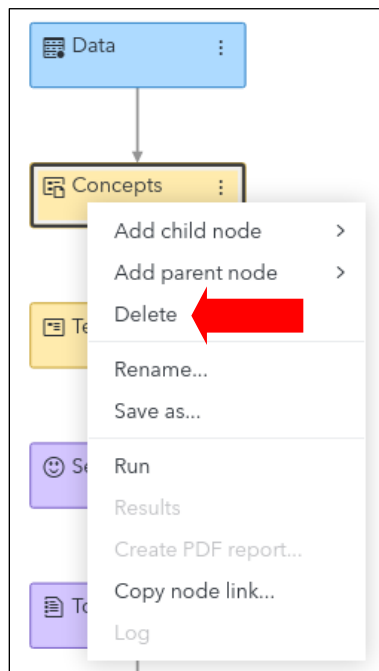
Variable Name	Type	Role	Display Variable
<input type="checkbox"/> __uniqueid__	Numeric	Key	
<input checked="" type="checkbox"/> DrugReport	Character	Text	✓
<input type="checkbox"/> EXTENSION	Character		
<input type="checkbox"/> ID	Character		
<input type="checkbox"/> NAME	Character		

16. Select **Pipelines** (in the upper left of the window and to the right of **Data**). The default Text Analytics pipeline appears. If you set up your project to be the Text Analytics type, you can access SAS Visual Text Analytics with Model Studio.

Default Text Analytics Pipeline

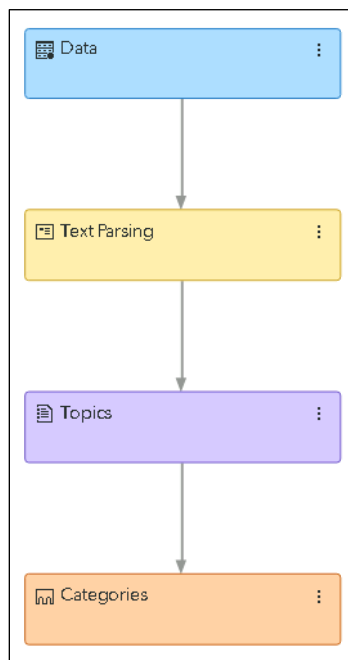


17. For this project, no concepts are used, and sentiment analysis is not relevant for this analysis on patient feedback. Delete the **Concepts** and **Sentiment** nodes. To delete a node, right-click the node or click the three vertical dots next to the name of the node, and then select **Delete**.



18. Click **Delete** again in the confirmation window that appears. The remaining nodes are reconnected automatically.

Modified Custom Text Analytics Pipeline



Defining Lists (Step 2 of 3)

Now you select optional term lists to include in your project. Start lists and stop lists enable you to control which terms are used or which terms are not used, respectively. In SAS Visual Text Analytics, you can use a start list or a stop list, but not both.

Note: A *start list* is a data set that contains a list of terms to include in the analysis results. If you use a start list, then only the terms that are included in that list are used in the analysis. A *stop list* is a data set that includes a list of terms to exclude from the analysis results, such as terms that contain little information or that are outside the realm of your analysis. A default stop list is provided for each of the languages that SAS Visual Text Analytics supports.

1. Select the **Text Parsing** node. The Text Parsing options appear to the right of the pipeline. If the options do not appear, click **Options**. The options menu is shown below.

The screenshot shows the 'Text Parsing' options panel. At the top, there's a title bar with a double arrow icon and three icons (search, play, and help). Below the title, there's a 'Description:' section with a text box containing 'Prepares text for terms analysis.' Underneath is a 'Minimum number of documents:' section with a slider ranging from 1 to 100, with major ticks at 1, 34, 67, and 100. The slider is currently set to 4. Below the slider is a 'Lists' section with a dropdown arrow. Under 'Lists', there's a checkbox labeled 'Specify a custom start or stop list'. Below that is a 'List type:' dropdown menu currently set to 'Stop list'. Under 'List type:', there are two sections: 'Start list:' and 'Stop list:'. Each has a 'Select a table' button and a 'Browse' button. At the bottom, there's a checkbox labeled 'Specify a synonym list' and a 'Synonym list:' field with a 'Browse' button.

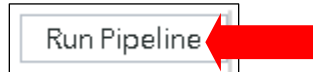
With the default settings, a default stop list is used for the language that is selected for the project. There is no default synonym table. Also, the **Minimum number of documents** property specifies the minimum number of documents that must contain a term before that term can be in the start list. The default value is 4, so if a term appears in only three documents, it is automatically assigned to the stop list, regardless of membership in either the specified start or stop list.

Note: A *synonym list* is a SAS data set that identifies pairs of terms that should be treated as a single term for analysis. The data set can include both a term and different forms of that term, including misspellings or abbreviations. For example, you can specify that the words *advert* and *advertising* should be treated as the term *advertisement*. You do not use a synonym list for this project, so do not select the **Specify a synonym list** check box.

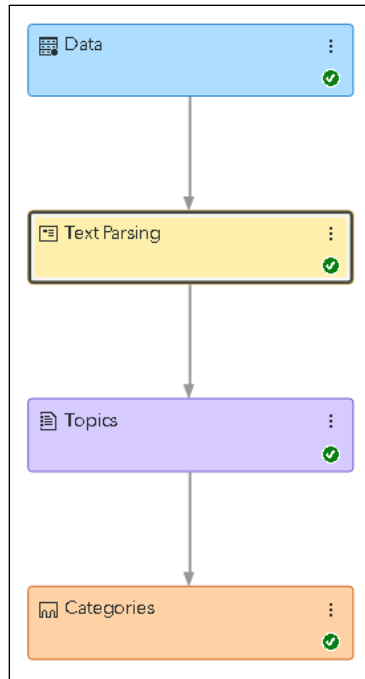
2. Default settings are used, so no further steps are required.

Running the Pipeline and Examining the Results (Step 3 of 3)

1. Run the entire pipeline. Right-click the last node and click **Run**. You can also click the **Run Pipeline** button.



When the run is complete, a message appears. If the run is successful, all the nodes display a green circle with a white check mark.



2. Right-click the **Text Parsing** node and click **Open**.

Model Studio - Build Models

Patient Feedback > Text Parsing - Manage Terms

Run node Close

Kept Terms (1386) Filter

Term	Role	Documents	Frequency
<input type="checkbox"/> not	ADV	658	1174
<input type="checkbox"/> > take	V	676	1104
<input type="checkbox"/> > depression	N	492	616
<input type="checkbox"/> > year	N	436	561
<input type="checkbox"/> > feel	V	379	540
<input type="checkbox"/> > work	V	378	488
<input type="checkbox"/> > drug	N	341	485

Dropped Terms (6320) Filter

Term	Role	Documents	Frequency
<input type="checkbox"/> i	PRO	1182	5576
<input type="checkbox"/> > be	V	1127	4156
<input type="checkbox"/> > have	V	1069	2849
<input type="checkbox"/> and	CONJ	1028	2746
<input type="checkbox"/> the	DET	903	2558
<input type="checkbox"/> to	PPOS	895	2383
<input type="checkbox"/> it	PRO	878	2082

Documents

All (1414) Matched Search

DrugReport

This medication made me gain 40 pounds it has been 2 years and I have only lost 10 pounds. Beware and watch your weight.

causing extreme anger, to the point that my family has become afraid of me. Doing things i would have considering doing before, like challenging the police, starting arguments, really, really wanting to beat the crap out of someone. Enough to the point that it scares me. But my doctor at this time refuses to take me off this medication. I am scared i will end up in jail for severely hurting someone.

The Kept Terms pane contains the start list. These 1,389 terms are used in the terms table, which will be used in subsequent nodes in the analysis. The Dropped Terms pane contains the

6,378 terms that will be ignored in the analysis. These dropped terms make up the stop list. All terms within the document collection are accounted for.

3. In the Kept Terms pane, select the check box next to the noun (N) form of **depression**.

Kept Terms (1386) <input type="text" value="Filter"/>				
	Term	Role	Documents	Frequency
<input type="checkbox"/>	not	ADV	658	1174
<input type="checkbox"/>	take	V	676	1104
<input checked="" type="checkbox"/>	depression	N	492	616
<input type="checkbox"/>	year	N	436	561

Note: In certain cases, using the filter at the top of the list can save time. It is not needed in this case because the term that we want appears at the top of the Kept Terms list.

4. Click **Show term map**.

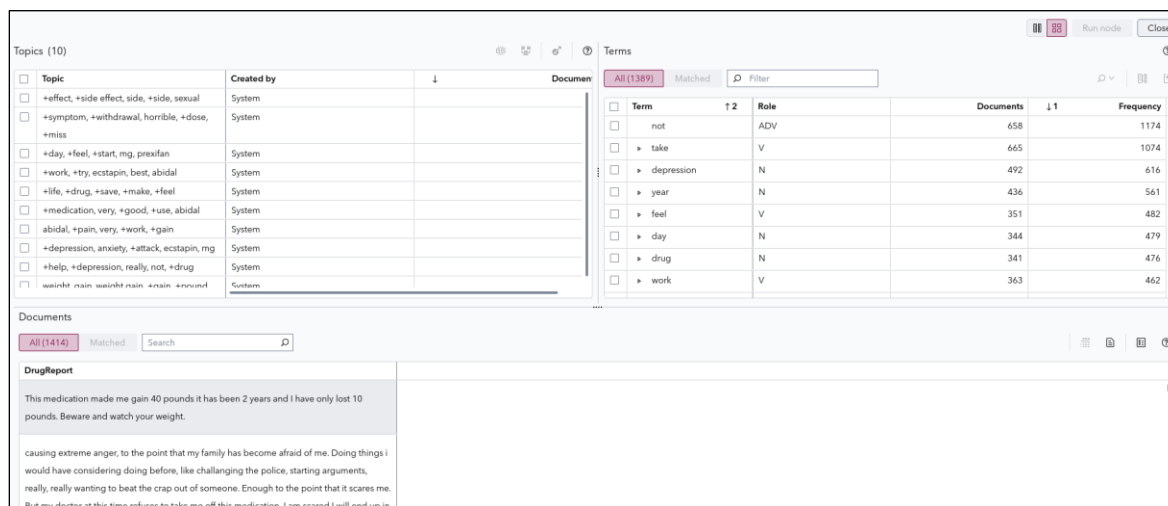
Note: Depending on your screen resolution, if you do not see the shortcut button for **Show term map**, click the **More options** button (the three vertical dots) next to the Filter window and then select **Show term map** from the drop-down menu.



A term map is useful for identifying associations between terms. The term map is most useful when it identifies associations that were previously unknown. Even when associations are known, a measure called **information gain** can be calculated to estimate the strength of the

association. The thickness of the connectors between terms visually displays the relative information gain. Thicker lines imply higher information gain, which suggests a stronger association. Multiple other terms are related to the term *depression*. Related terms with the strongest associations are *anxiety*, *major depression*, and *suffer*.

- Close the term map by clicking **Close** in the upper right corner of the window.
- Close the Text Parsing window.
- Right-click the **Topics** node and click **Open**. You can see that 10 machine-generated topics are generated by the Topics node.



The screenshot shows the SAS Visual Text Analytics interface. The 'Topics (10)' window is open, displaying a list of 10 topics. Each topic has a checkbox, a topic description, and a 'Created by' field. The 'Documents' column shows the number of documents associated with each topic. The 'Terms' window is also visible, showing a list of terms and their roles.

Topic	Created by	Documents
<input type="checkbox"/> +effect, +side effect, side, +side, sexual	System	238
<input type="checkbox"/> +symptom, +withdrawal, horrible, +dose, +miss	System	220
<input type="checkbox"/> +day, +feel, +start, mg, prexifan	System	202
<input type="checkbox"/> +work, +try, ecstapin, best, abidal	System	200
<input type="checkbox"/> +life, +drug, +save, +make, +feel	System	189
<input type="checkbox"/> +medication, very, +good, +use, abidal	System	188
<input type="checkbox"/> abidal, +pain, very, +work, +gain	System	186
<input type="checkbox"/> +depression, anxiety, +attack, ecstapin, mg	System	158
<input type="checkbox"/> +help, +depression, really, not, +drug	System	140
<input type="checkbox"/> weight, gain, weight gain, +gain, +pound	System	117

The 10 machine-generated topics are shown below.

Topics (10)			
<input type="checkbox"/> Topic	Created by	↓	Documents
<input type="checkbox"/> +effect, +side effect, side, +side, sexual	System		238
<input type="checkbox"/> +symptom, +withdrawal, horrible, +dose, +miss	System		220
<input type="checkbox"/> +day, +feel, +start, mg, prexifan	System		202
<input type="checkbox"/> +work, +try, ecstapin, best, abidal	System		200
<input type="checkbox"/> +life, +drug, +save, +make, +feel	System		189
<input type="checkbox"/> +medication, very, +good, +use, abidal	System		188
<input type="checkbox"/> abidal, +pain, very, +work, +gain	System		186
<input type="checkbox"/> +depression, anxiety, +attack, ecstapin, mg	System		158
<input type="checkbox"/> +help, +depression, really, not, +drug	System		140
<input type="checkbox"/> weight, gain, weight gain, +gain, +pound	System		117

- Select the **+life, +drug, +save, +make, +feel** check box. This topic could be related to positive patient recovery, and discovering such themes was stated as the primary goal of the analysis.

Topics are identified using the five terms that have the largest relevancy score within that topic.

9. In the Terms window, click **Matched**.

Terms					
All (1389)		Matched(98 of 1389)		Filter	
<input type="checkbox"/>	Term	↑ 2	↓ 1	Relevancy	Documents
<input type="checkbox"/>	▶ life			0.442	N
<input type="checkbox"/>	▶ drug			0.237	N
<input type="checkbox"/>	▶ save			0.221	V
<input type="checkbox"/>	▶ make			0.211	V
<input type="checkbox"/>	▶ feel			0.166	V

In the start list for this collection, 98 of the 1,389 terms have relevancy weights that are greater than the term cutoff that is specified in the Topics settings window. Terms are sorted by descending relevancy score. As expected, the top five relevancy scores correspond to the terms that are used to name the topic. Terms with a plus sign have stemmed terms that are indicated in the Terms table by a right arrow. If you click the arrow, the stemmed terms are displayed, and if a synonym list is used, the synonyms are also displayed.




10. Click the right arrow next to the term **life** in the Terms table to see its stemmed versions.


▶ life
life
lives

11. Examine the Documents table for the selected topic. Click **Matched**.

Documents	
All (1414)	Matched (189 of 1414)
Search	
DrugReport	
This drug saved my life !!	
life changingfor the good	
I feel so forutnate to have had Abidal prescribed to me. It has made a world of difference	
in my life , and I feel so much better .	
easy to use and made my life so much better :)	

Out of the 1,414 total documents, 189 documents are identified as belonging to the topic, based on the document relevancy score cutoff that is specified in the Topics node settings. Any of the 98 terms that are used to identify the topic that appears in the document are highlighted. Documents are sorted by the descending document relevancy score.

Three actions are available for topics: Split topics , Merge topics , and Add topics as categories .

12. Click **Add topics as categories** . Adding a topic as a category is also called *promoting* the topic. A message briefly appears, and it indicates that a topic was added as a category, and the Add topics as categories icon appears next to the promoted topic's name

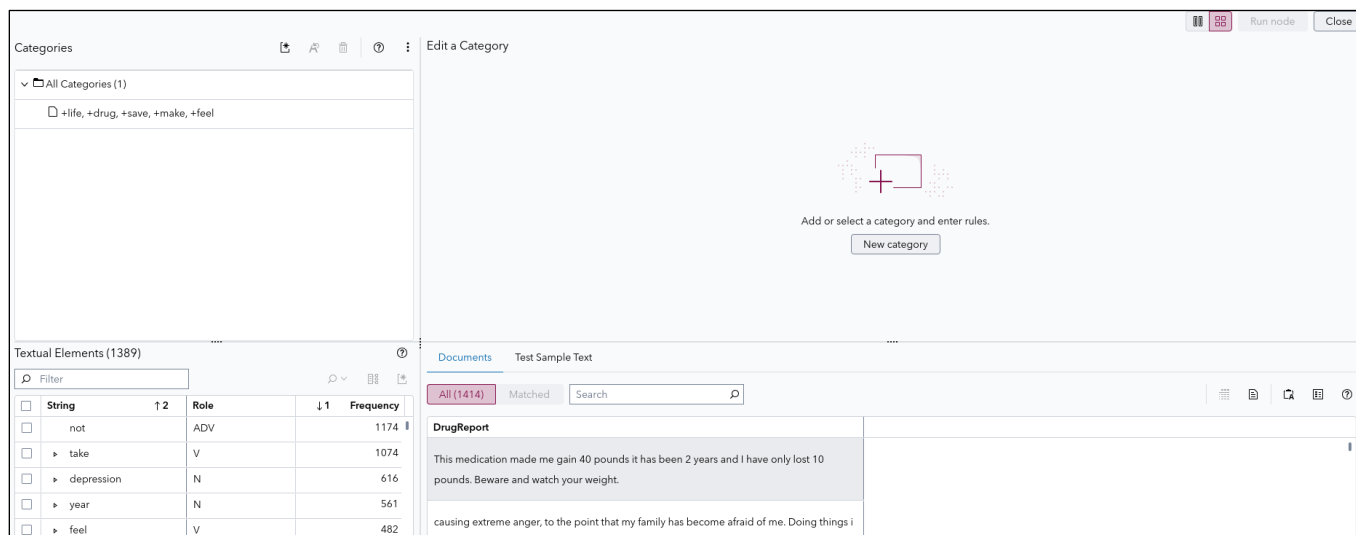
<input type="checkbox"/>	+work, +try, ecstapin, best, abidal	System
<input checked="" type="checkbox"/>	+life, +drug, +save, +make, +feel	System
<input type="checkbox"/>	+medication, very, +good, +use, abidal	System

13. Close the Topics node window.

In the pipeline, the green circle around the check mark next to the Categories node has been replaced by a gray circle. This indicates that the node needs to be rerun to accommodate the new information about a topic promoted to a category.

14. Run the **Categories** node. Because the Categories node is the last node in the pipeline, all out-of-date nodes preceding the Categories node are run. The promoted topic is added as a category, and category rules are derived for the topic.

15. Open the **Categories** node.

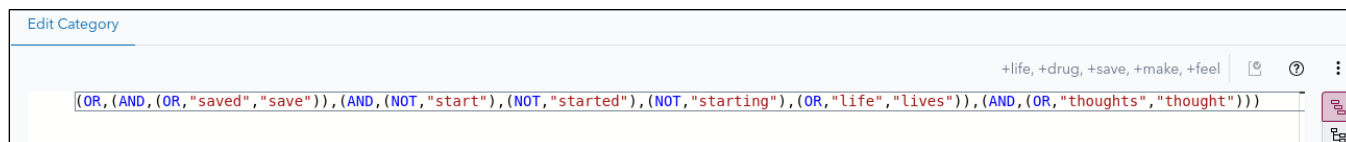


	String	Role	Frequency
<input type="checkbox"/>	not	ADV	1174
<input type="checkbox"/>	take	V	1074
<input type="checkbox"/>	depression	N	616
<input type="checkbox"/>	year	N	561
<input type="checkbox"/>	feel	V	482

Only a single categorical entry is present. It is related to the binary topic category that was promoted from the Topics node. If variables on the Data tab had an assigned role of Category, the variables along with their categorical levels would also appear.

16. Select **+life, +drug, +save, +make, +feel**.

The category Boolean rule shown below was generated by SAS Visual Text Analytics software for the promoted topic.





```
(OR, (AND, (OR, "saved", "save")), (AND, (NOT, "start"), (NOT, "started"), (NOT, "starting")), (OR, "life", "lives")), (AND, (OR, "thoughts", "thought")))
```


17. To the right of the Edit Category window, click **Tree view**  to see the rule tree hierarchy that can help you understand the rule.

▼ OR
▼ AND
▼ OR
"saved"
"save"
▼ AND
▼ NOT
"start"

18. In the Documents window, click **Matched**. The matched documents results associated with **+life**, **+drug**, **+save**, **+make**, **+feel** appear.

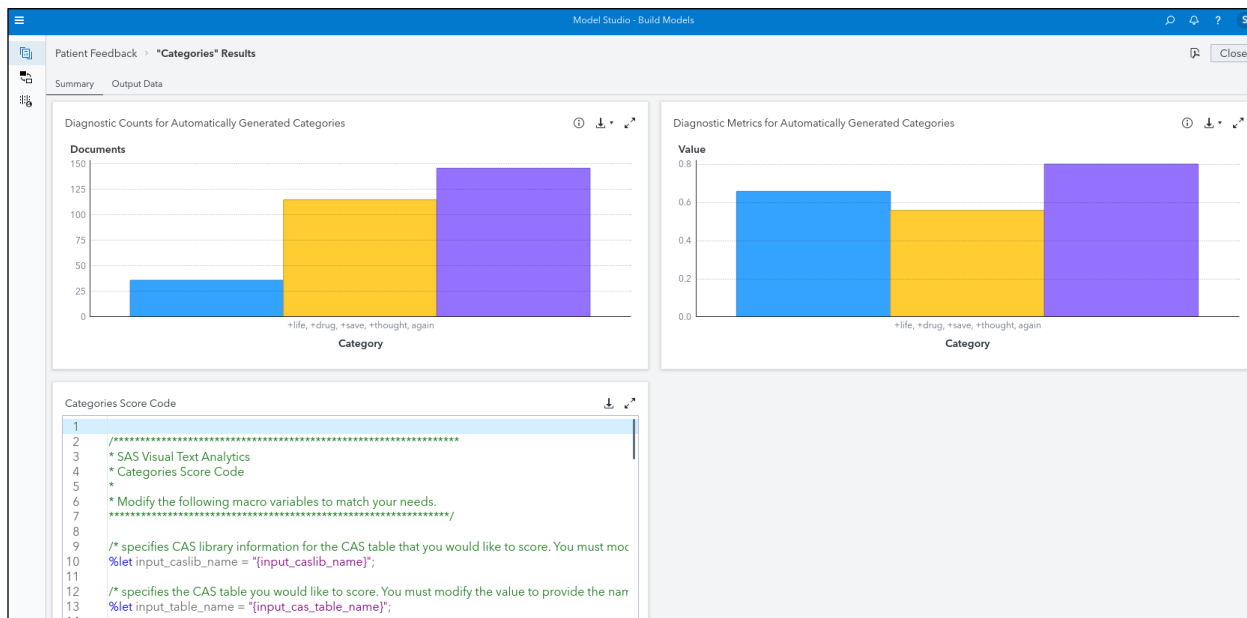
Documents Test Sample Text	
All (1414)	Matched (225 of 1414) Search 
DrugReport	Relevancy
hey everyone ecstapin has saved my life after my son was born i had severe postpartum not baby blues i had very bad thoughts about my baby and me my mom was taking ecstapin so i asked a psy,dr if they would put me on ecstapin too so she did after already hadtaken lots of different ones it worked i went threw pp depression for over 2 yrs off and on i have to say if u dont really need it dont take it but if your bad off it can save your life !!! i would like to get off of it now but if i miss 2 days i get to crying and depressed again so	6.000 

A total of 225 documents satisfy the category definition for this topic. Earlier, you saw that a total of 189 documents exceeded the relevancy score cutoff in the Topics node for the topic. At least 36 documents were misclassified by the Category node.

Examining these documents could cause you to change the relevancy cutoff (the Topics node) or to modify the category Boolean script (the Categories node).

19. Close the Categories window.

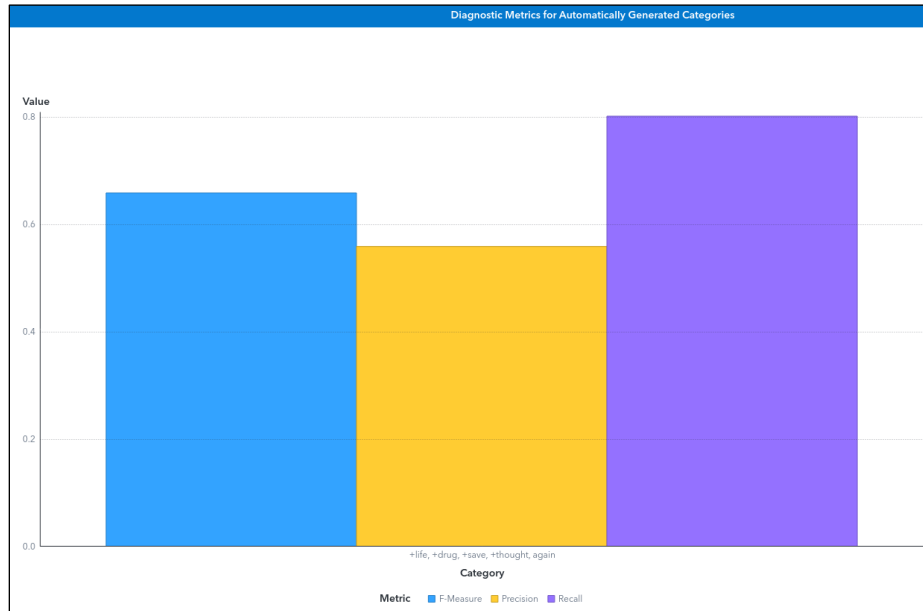
20. Right-click the **Categories** node and select **Results**. (If a gray circle still appears on the Categories node and the Results option is not available from the pop-up menu, run the node again and then open the results.)



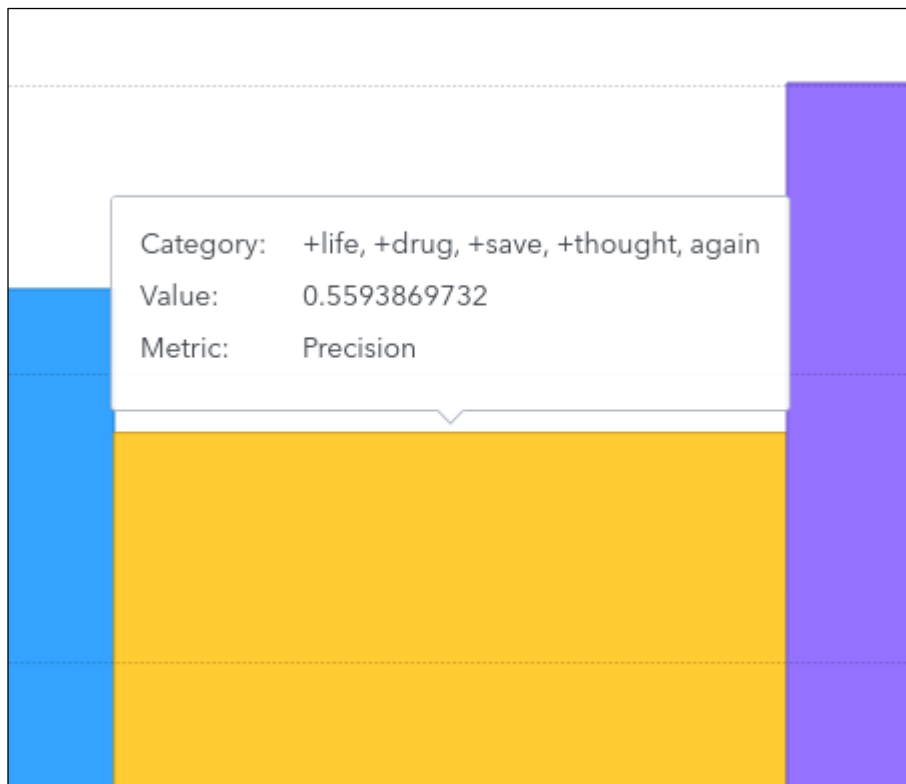
Three panes appear: Diagnostic Counts, Diagnostic Metrics, and Categories Score Code. The Diagnostic Counts pane shows bars representing false negatives, false positives, and true positives. True negatives are not shown because for the typical situation where negatives outnumber positives, the number of true negatives tends to dwarf the other bars. The following plots show visual representation of misclassified and correctly classified documents to illustrate why plotting true negatives is ill advised.

The Diagnostic Metrics pane shows a bar chart with bars for F-Measure (F1 statistic), precision, and recall.

21. Expand the diagnostics metrics pane by clicking **Expand**  .



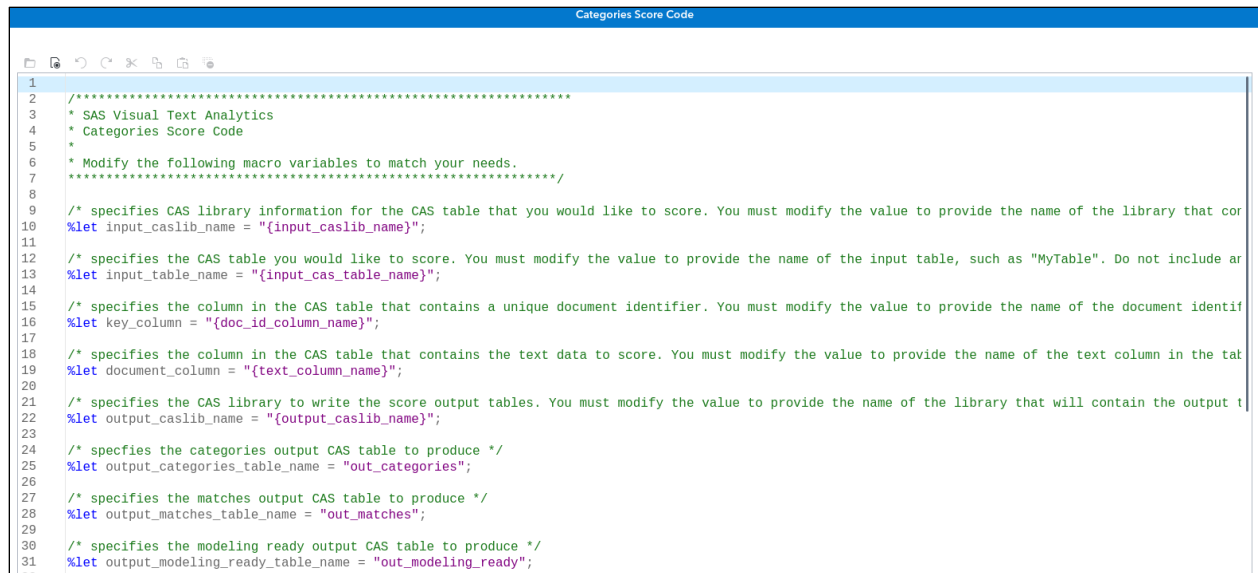
If you position the cursor on a bar, you see the numeric value of the statistic that is represented by the bar. The pop-up statistic for precision appears in the following screenshot:



The precision for the category rule based on the promoted topic **+life, +drug, +save, +make, +feel** is 0.5511.

22. Restore the Diagnostic Metrics plot by clicking **Close** (the X).

23. Expand the Categories Score Code window.



```

1
2
3 /* SAS Visual Text Analytics
4  * Categories Score Code
5  *
6  * Modify the following macro variables to match your needs.
7  */
8
9 /* specifies CAS library information for the CAS table that you would like to score. You must modify the value to provide the name of the library that cor
10 %let input_caslib_name = "{input_caslib_name}";
11
12 /* specifies the CAS table you would like to score. You must modify the value to provide the name of the input table, such as "MyTable". Do not include ar
13 %let input_table_name = "{input_cas_table_name}";
14
15 /* specifies the column in the CAS table that contains a unique document identifier. You must modify the value to provide the name of the document identif
16 %let key_column = "{doc_id_column_name}";
17
18 /* specifies the column in the CAS table that contains the text data to score. You must modify the value to provide the name of the text column in the tat
19 %let document_column = "{text_column_name}";
20
21 /* specifies the CAS library to write the score output tables. You must modify the value to provide the name of the library that will contain the output t
22 %let output_caslib_name = "{output_caslib_name}";
23
24 /* specifies the categories output CAS table to produce */
25 %let output_categories_table_name = "out_categories";
26
27 /* specifies the matches output CAS table to produce */
28 %let output_matches_table_name = "out_matches";
29
30 /* specifies the modeling ready output CAS table to produce */
31 %let output_modeling_ready_table_name = "out_modeling_ready";

```

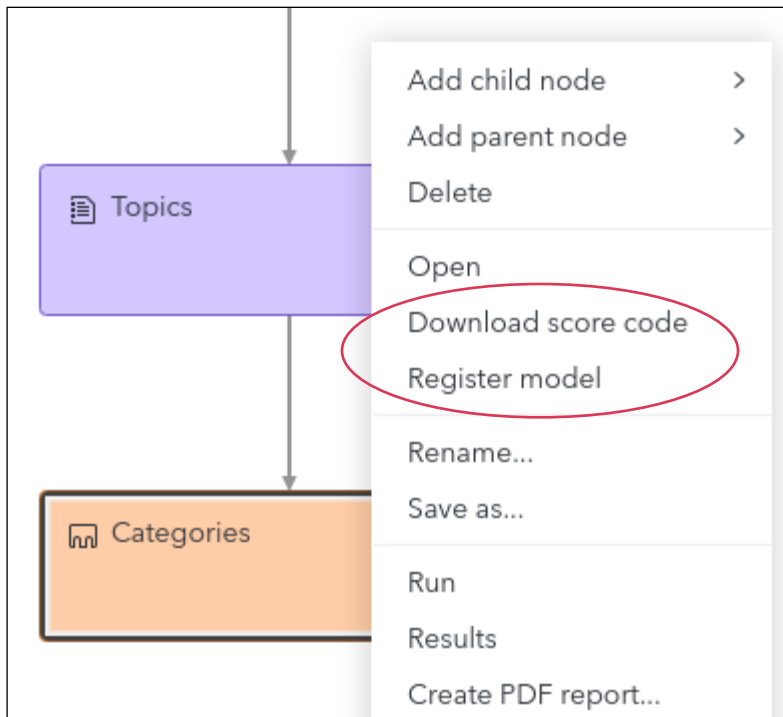
This score code is automatically generated and is quite simple to use. The code uses macro variables within the first several lines to take in key information for scoring such as input and output libraries, the unique document identifier (ID), and the table of documents to be scored. This portion of the code is easy to edit by simply replacing items inside braces with the relevant information needed for scoring.

Running the score code produces three output tables: **out_categories**, **out_matches**, and **out_modeling_ready**. Typically, these three output tables should be renamed to be specific for the analysis at hand and to avoid ambiguities with scored output from other Visual Text Analytics projects.

We show in the next steps how the score code can be downloaded as a ZIP file or how the model can be registered directly into SAS Model Manager for scoring.

24. Restore the Score Code window and close the results.

25. Right-click the **Categories** node and observe the pop-up window that appears.



By selecting **Download score code**, the web browser creates a ZIP file of the score code, which makes the code easily transportable depending on the scoring environment. For the Categories node, the download file is generically named CategoriesScoreCode.zip. (A best practice is to rename this ZIP file to be specific to the analysis at hand.) Within the Categories node zipped score code file is the SAS DATA step score code. This ScoreCode.sas file has preliminary commented entries to guide you about how to complete the program for scoring a specific data set on your system. Use SAS Studio to edit and run the program. The ScoreCode.sas file is the same score code observed earlier that is visible within the results window of the node.

Score code is available for concepts, sentiment, topics, and categories. You can score any document collection that has a document variable with the same name as the document variable that is used to generate the score code.

Some of the nodes in the Visual Text Analytics pipeline that produce score code produce both DATA step and ASTORE score code files.

By selecting **Register model**, the text analytic model in the form of score code is registered to SAS Model Manager for deployment and other ModelOps activities.

For the sake of brevity, we will not run the score code in this demonstration.

End of Demonstration



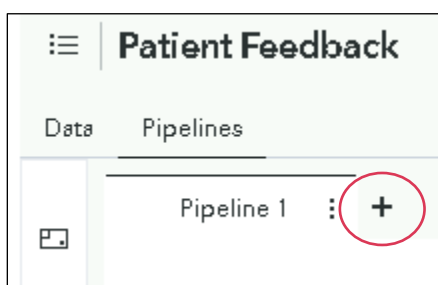
Using Predefined and Custom Concepts in a SAS Visual Text Analytics Project

This demonstration illustrates the advanced functionality of SAS Visual Text Analytics and specifically focuses on the Concepts node. Both predefined and custom concepts are considered.

This demonstration continues to use the **drug_reports** data and is performed within the Patient Feedback project created earlier. The primary goal is to extract specific drug dosages from the patient feedback documents.

Extracting Drug Dosages using Concepts

1. Make sure that the **Patient Feedback** project created in the earlier demonstration is opened and on the Pipelines tab.
2. Click the **Add new pipeline (+)** button found next to the tab for Pipeline 1.



3. Enter **Explore Concepts** as the name for the pipeline. The default text analytics pipeline template will be used.

New Pipeline

Name: *

Description:

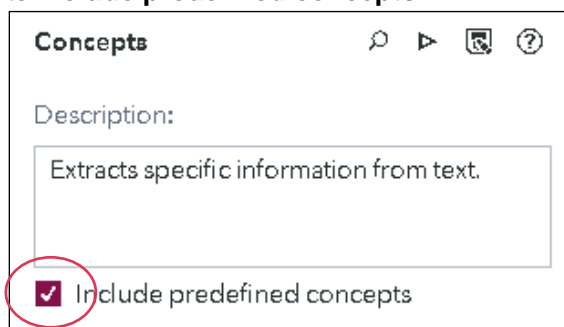
Template:

Language:

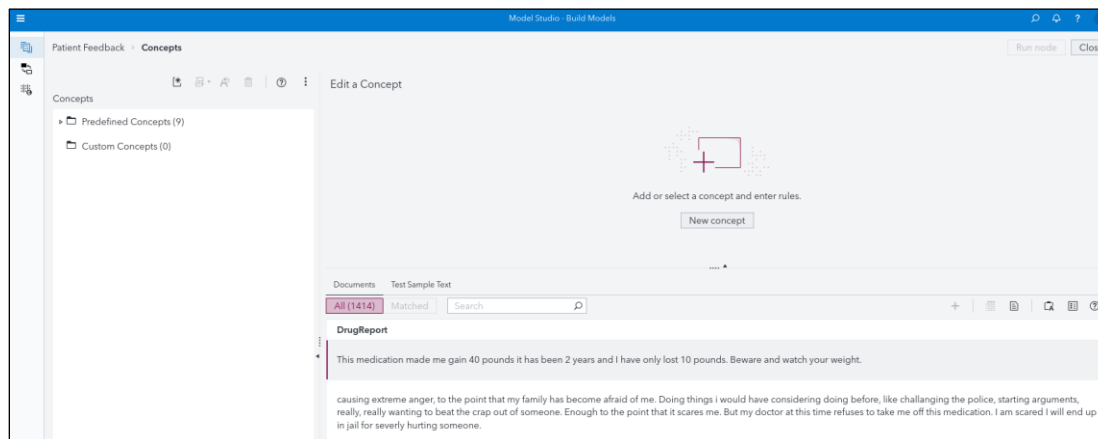
4. Click **Browse** next to Template. A full list of all available Visual Text Analytics templates is shown. Select the template with the name **Text Analytics: Generate Concepts, Topics, and Categories**. This is the default template.

Browse Templates			
Filter			
Template Name	Description	Owner	Last Modified
Text Analytics: Assisted Concept Rule Creation	Use Textual Elements to quickly generate custom concept rules.	SAS Pipeline	March 4, 2025 at 09:14:32 PM
Text Analytics: Data Access	Contains a single Data node.	SAS Pipeline	March 4, 2025 at 09:14:25 PM
Text Analytics: Generate Concepts, Topics, and Categories	Text Analytics pipeline for model generation with Concepts, Text Parsing, Sentiment, Topics, Categories.	SAS Pipeline	March 4, 2025 at 09:14:11 PM
Text Analytics: Topic Discovery	Text Analytics pipeline that uses text parsing and machine learning to discover topics.	SAS Pipeline	March 4, 2025 at 09:14:34 PM

- Click **OK**. Click **Save** in the New Pipeline window.
- Select the **Concepts** node and observe the properties pane on the right. There is a single property for the node, which indicates to include predefined concepts. Select the check box next to **Include predefined concepts**.

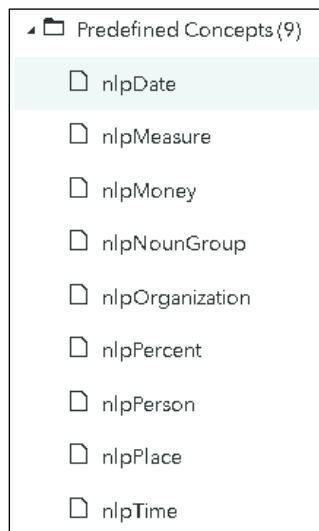


- Right-click the **Concepts** node and select **Run**.
- When the run is complete, right-click the **Concepts** node and select **Open**.



In the Concepts pane (upper left corner), notice that nine predefined concepts have been used.

- Click the arrow to the left of **Predefined Concepts (9)**. The nine predefined concepts are shown, each with an **nlp** prefix. Information within each of the concepts is extracted using natural language processing.



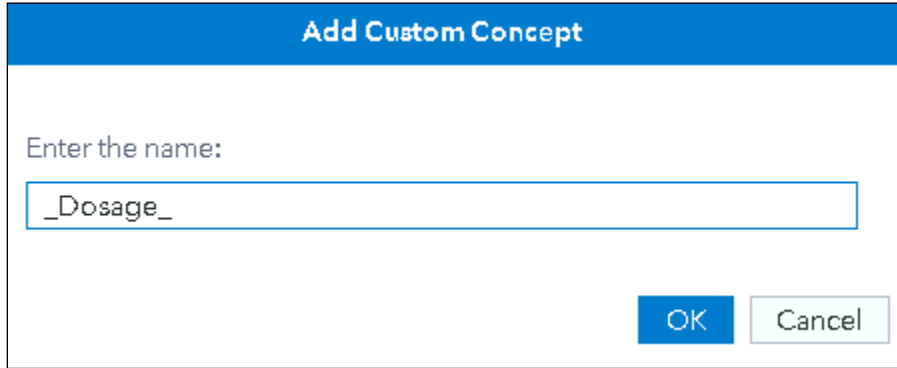
10. Select **nlpMeasure**. This concept relates to measures. The LITI (language interpretation for textual information) code used to define the nlpMeasure concept is not displayed in the Edit Concept window. The software does not reveal the code for any of the predefined concepts.
11. The Documents pane is in the lower right corner of the display. In the documents pane, click **Matched**.


Documents Test Sample Text	
All (1414) Matched (784 of 1414) Search	
DrugReport	Fact Mat...
...made me gain 40 pounds it has been 2 years and I have only lost 10 pounds . Beware and watch your weight.	0
...to my ecstapin(225mg) due to unrelenting depression.I had lost my sisiter mom within 8 months and although I had benn on an antidepressant for a long time before I grieved but still couldn't get over the depression. Within 3-54 daysboth I and my husband noticed a significant improvement. I felt better more like me than I had in years. Only thing I notice was I sometimes mix up words Anybody else do that? Not bad enough for me to stop med tho.	0
...been on ecstapin 150mg until i started having breakthroughs.My doctor has started me on abidal 40mg daily. I have the dizziness,but my mood is awesome. I do not know if taking more will make the dizziness go away or when will it?? That is my only side effect, I think? For the one person who has the rash and other side effects...you are allergic!!! Stop taking it.	0

Note that the nlpMeasure concept finds numerical measurements such as *weight* and *time* (the blue highlighted terms within the documents). It also discovers dosage amounts such as **225mg** found in the second document shown above. The nlpMeasure concept finds measures in 784 of the 1,414 total documents. Some documents might contain more than a single measure.

The goal of the analysis is to extract drug dosages. The nlpMeasure predefined concept found drug dosages, but other numeric measures are also extracted. This predefined measure does not address the analysis goal. Because the LITI code to discover the measures is not shown nor can it be edited, we must create a custom concept to extract only dosages.

12. Select **Custom Concepts (0)** on the left. Click the **New Concept** button in the Edit a Concept pane. A data entry window appears. Enter the name **_Dosage_** for the new custom concept.



Note: You can also create a custom concept by clicking the **New concept** shortcut button .

When selecting a concept name, you want to avoid ambiguities and unexpected results. See the “Create a Custom Concept” subsection in Chapter 6 in *SAS Visual Text Analytics: User’s Guide*. For example, avoid using names that are actual words that might appear in the term table. In this case, we have used the underscores (`_`) to differentiate the concept name from the term *dosage* that might appear within documents.

13. Click **OK**. The `_Dosage_` concept name is highlighted in the Concepts pane on the left, and the Edit Concept pane is now active.
14. Enter the following rules in the Edit Concept window. The code call also be copied from the file **Drug_Dosage.txt** located at **workshop > SIWTAS_Text**.

```
REGEX:[0-9]+[\.][0-9]+\s?mg\.?
REGEX:\d+\s?mg\.?
```

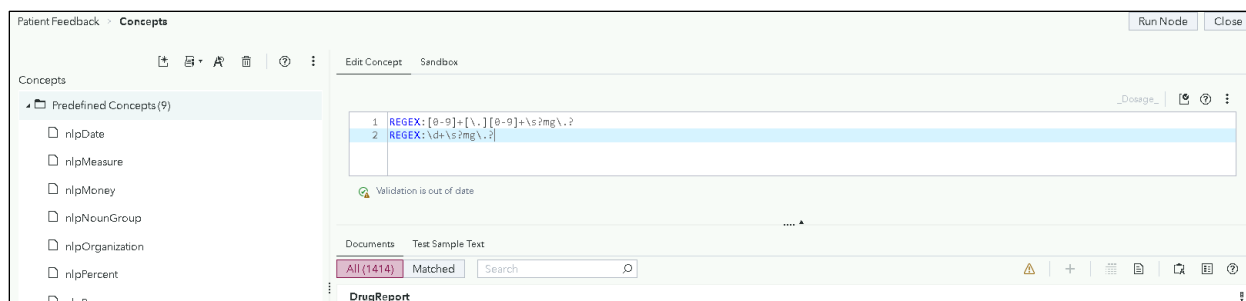
Although details about writing LITI code are beyond the scope of this workshop, an explanation of the first regular expression is as follows:

- a) `[0-9]`: match any single digit
- b) `+`: match the previous character one or more times (The previous character is a digit.)
- c) `[\.]`: match a period (decimal point). Note that the square brackets are not required.
- d) `[0-9]`: match any single digit
- e) `+`: match the previous character one or more times (The previous character is a digit.)
- f) `\s`: match any whitespace character (for example, space, tab)
- g) `?`: make the previous character optional (The previous character is a whitespace character.)
- h) `mg`: match the letters *mg*
- i) `\.`: match a period (signifying that *mg* is an abbreviation for milligrams)
- j) `?`: make the previous character optional (The previous character is a period.)


The second line of LITI code is similar to the first except that it does not look for decimal values.

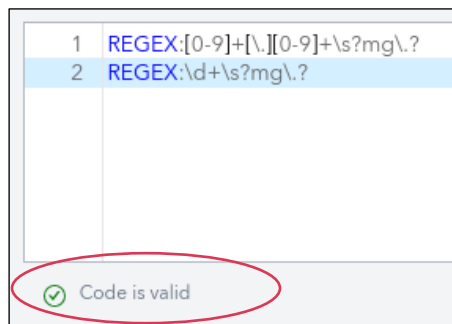
Note: There are additional LITI rules that can do complex content categorization and concept matching beyond what we are showing with this simple REGEX example. The purpose here is to provide a quick sample of what is possible when working with concepts in Visual Text Analytics.

15. After entering the LITI code, the Concepts window appears as follows:



Several steps are still required to use the new `_Dosage_` concept.

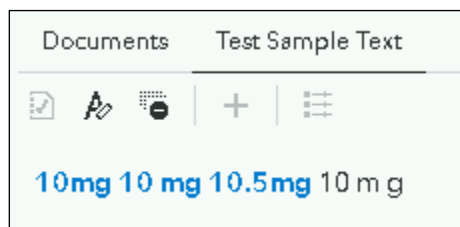
16. The message **Validation is out of date** means that you should validate the script before you submit it. To do so, click the **Validate Rules** shortcut button  in the upper right of the Edit Concept window. If you are successful, you see a new message, **Code is valid**.



17. Because document collections can be large, you might want to test the script on example text rather than run it on the entire data set. You can use the Test Sample Text feature to supply a few documents (or just a few lines of text) to test the rules that you supplied. Click the **Test Sample Text** tab (next to the Documents tab) and enter the following text:

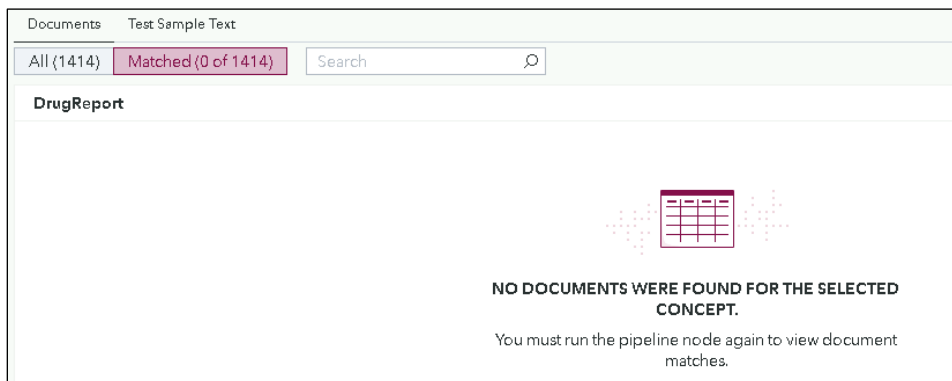
10mg 10 mg 10.5mg 10 m g

18. Click the **Test Sample Text**  shortcut button.



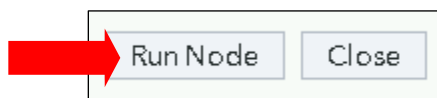
Three of the four dosages are matched. The last dosage is not matched due to the space between the *m* and the *g*.

19. Click the **Documents** tab and then select **Matched** in the Documents pane.



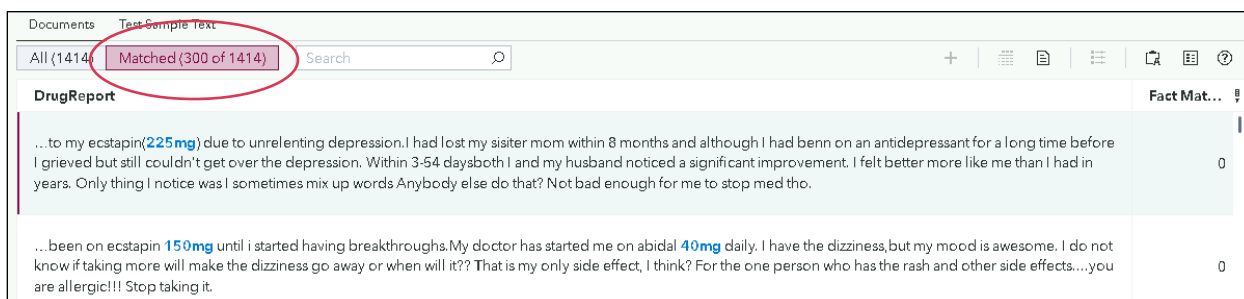
Note that the window shows that 0 of 1,414 documents matched. We know that the LITI code was validated and tested, so why were no documents found with matches for the `_Dosage_` concept? The Concepts node must be run.

20. Click the **Run Node** button in the upper right corner of the window.

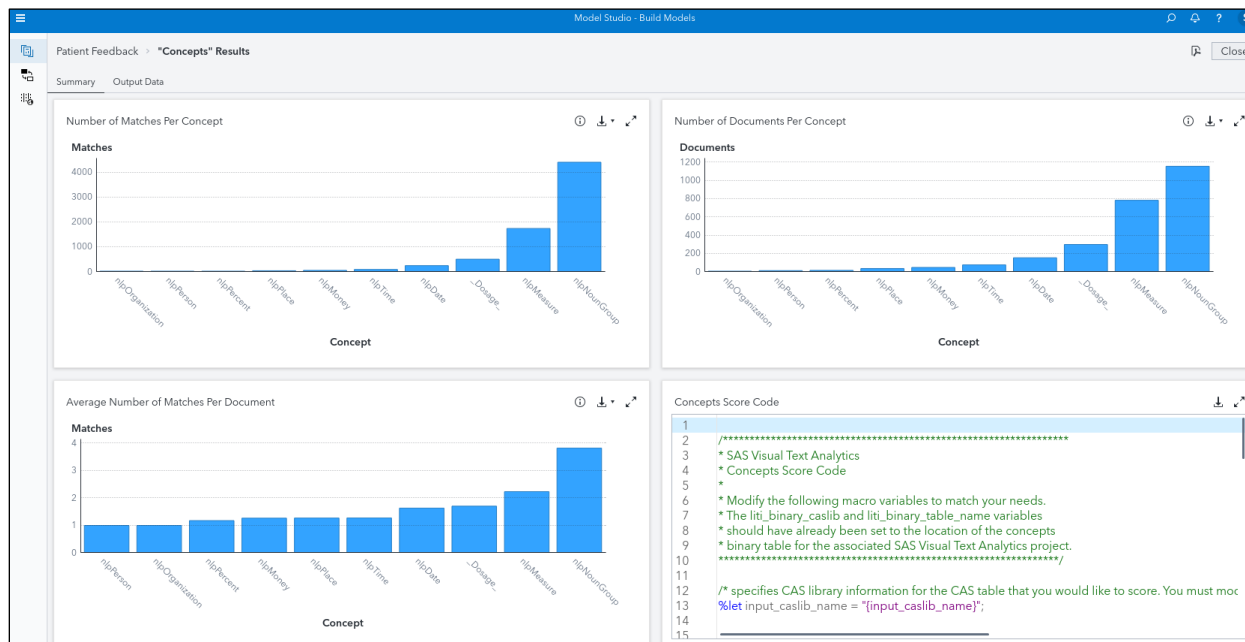


Note: You can also close the Concepts window and run the node as usual directly from the pipeline.

You now see that there were 300 total documents that contained matches to the `_Dosage_` concept.



21. Close the Concepts window.
22. Right-click the **Concepts** node and select **Results**. A summary of the results of the Concepts node is shown.



The Number of Matches Per Concept plot shows the total number of matches for each concept, whether predefined or custom. The plot rank orders the results from fewest matches to most. Keep in mind that a single document could contain several matches per concept. The concept `nlpNounGroup` has the greatest number of matches at 4649. The custom `_Dosage_` concept has the third most matches with 511.

The Number of Documents Per Concept plot shows the total number of documents in which each concept is found. The concept `nlpNounGroup` appears in more document than the other concepts. It appears in 1,165 documents (out of the 1,414 total). The `_Dosage_` concept is again in third place (from the concept with the most), appearing in 300 documents. This is the same number of matches that we saw earlier when creating the custom concept.

The Average Number of Matches Per Document plot shows the average number of appearances within a single document for each concept. These values are calculated by taking the total number of matches for a concept and dividing by the total number of documents in which the concept appears. `nlpNounGroup` has the largest average at 3.99 matches per document. `_Dosage_` is in third place with an average of 1.7.

The Concepts Score Code window provides the SAS DATA step score code for the Concepts node.

To score new documents, the Concepts score code could be accessed and used in multiple ways, similar to what was discussed in the prior demonstration for the Categories node. The score code could be copied directly from the Concepts Score Code window and pasted into a desired location, a ZIP file could be downloaded from the pipeline, and the score code can be uploaded as desired, or the model could be registered directly to SAS Model Manager. As stated before, macro variables embedded within the score code would need to be updated with the correct information for your specific analysis.

For the sake of brevity, we will not run the score code for this demonstration.

23. Close the results of the Concepts node.

24. The goal of extracting dosage amounts from the patient feedback documents has been achieved. The remainder of this demonstration is to illustrate how the use of concepts at the beginning of a pipeline can affect the terms table and, thus, affect the results of the analysis.

Illustrating That Using Concepts Affects the Analysis Results

Because the Concepts node comes before the Text Parsing node in the pipeline, concepts can appear in the terms table. As stated above, whether concepts are used can affect the terms table and, thus, affect the analysis. Let's explore this.

1. Right-click and select **Run** on the Text Parsing node.
2. When the run is complete, right-click the **Text Parsing** node and select **Open**.

Term	Role	Documents	Frequency
<input type="checkbox"/> not	ADV	658	1174
<input type="checkbox"/> > take	V	676	1104
<input type="checkbox"/> > depression	N	492	616
<input type="checkbox"/> > year	N	436	561
<input type="checkbox"/> > feel	V	379	540
<input type="checkbox"/> > work	V	378	488
<input type="checkbox"/> > drug	N	341	485
<input type="checkbox"/> > day	N	344	479
<input type="checkbox"/> > medication	N	304	446

Term	Role	Documents	Frequency
<input type="checkbox"/> i	PRO	1182	5576
<input type="checkbox"/> > be	V	1127	4156
<input type="checkbox"/> > have	V	1069	2849
<input type="checkbox"/> and	CONJ	1028	2746
<input type="checkbox"/> the	DET	903	2558
<input type="checkbox"/> to	PPOS	895	2383
<input type="checkbox"/> it	PRO	878	2082
<input type="checkbox"/> a	DET	792	1655
<input type="checkbox"/> my	DET	809	1551

Notice that there are now 1,469 kept terms. When no concepts were used, there were 1,389 kept terms. Using concepts in the analysis has affected the terms table because it will now contain concepts.

3. Scroll down in the Kept Terms pane to look for a term with a role of a Concept. Below you see that *side effect* appears with a role of nlpNounGroup.

Kept Terms (1469) <input type="text" value="Filter"/>					
<input type="checkbox"/>	Term	↑ 2	Role	Documents	↓ 1 Frequency
<input type="checkbox"/>	▶ help		V	294	341
<input type="checkbox"/>	very		ADV	253	332
<input type="checkbox"/>	▶ side effect		nlpNounGroup	268	329
<input type="checkbox"/>	side		A	271	318

4. Close the Text Parsing window.
5. Right-click the **Topics** node and select **Run**.
6. When the run is complete, right-click the **Text Parsing** node and select **Open**.

Topics (10)			
<input type="checkbox"/>	Topic	Created by	↓ Documents
<input type="checkbox"/>	+effect, side, +side effect, +side, sexual	System	23
<input type="checkbox"/>	+day, +feel, mg, +start, +night	System	22
<input type="checkbox"/>	+symptom, +withdrawal, horrible, +dose, +drug	System	21
<input type="checkbox"/>	+work, +try, not, ecstapin, best	System	19
<input type="checkbox"/>	+medication, very, +good, +use, anxiety	System	19
<input type="checkbox"/>	+life, +save, +drug, +feel, +make	System	18
<input type="checkbox"/>	abidal, +pain, very, +work, +week	System	18
<input type="checkbox"/>	+help, +depression, anxiety, really, +drug	System	16
<input type="checkbox"/>	mg, +depression, ecstapin, anxiety, +attack	System	14
<input type="checkbox"/>	weight, gain, weight gain, +gain, +pound	System	11

There are still 10 system-generated topics, but they might have changed from what was seen in the previous demonstration.

Recall that the goal of the first demonstration was to look for themes within the patient feedback documents, specifically looking for documents that might contain a theme of positive patient recovery. The topic discovered earlier was based on the five key terms **+life**, **+drug**, **+save**, **+make**, **+feel**, and that topic appeared in 189 documents.

In the new analysis, that topic does not exist, but it likely has been replaced with the topic **+life**, **+save**, **+drug**, **+feel**, **+make**.

7. Select the topic **+life, +save, +drug, +feel, +make**. In the Document pane, select **Matched**.



The screenshot displays the 'Documents' section of the SAS Visual Text Analytics interface. At the top, there are two buttons: 'All (1414)' and 'Matched (188 of 1414)'. The 'Matched' button is highlighted with a red oval. To the right of these buttons is a search bar with the text 'Search' and a magnifying glass icon. Below the buttons, a table lists documents. The first document is titled 'DrugReport' and contains several paragraphs of text with highlighted words in blue, such as 'drug', 'life', 'feel', 'better', 'life changing', 'good', 'easy', 'made', 'suffered', 'effective', and 'drug'.

Document	Matched
DrugReport	
This drug saved my life !!	
I feel so forutnate to have had Abidal prescribed to me. It has made a world of difference in my life , and I feel so much better .	
life changingfor the good	
easy to use and made my life so much better :)	
I suffered most of my life with treatment-resistant depression before taking Efexor. This is the only drug that has been effective for me.	

Notice that the new topic appears in 188 documents.

End of Demonstration