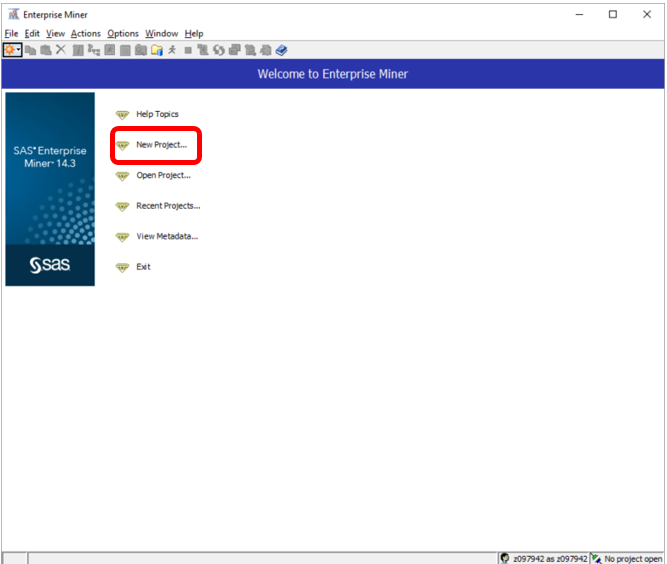
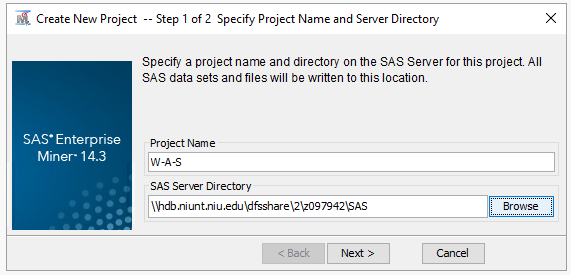
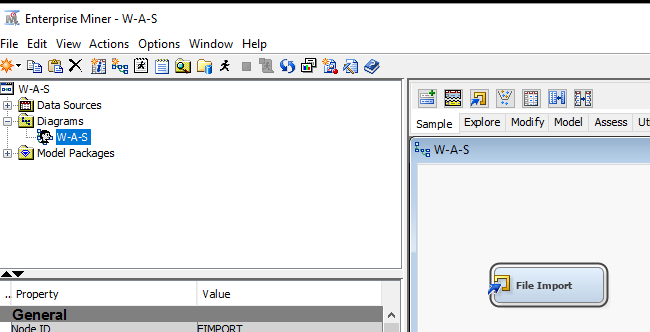
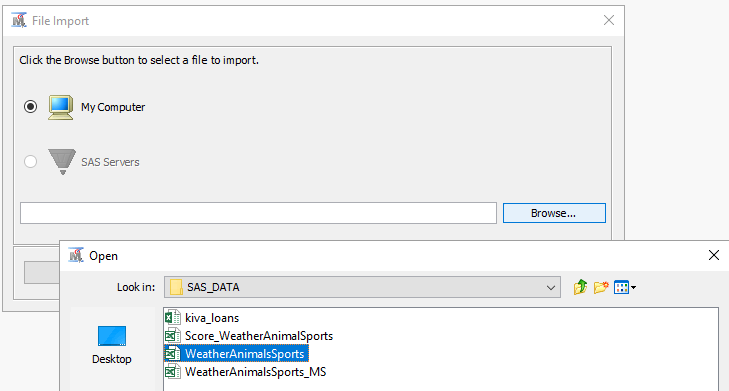
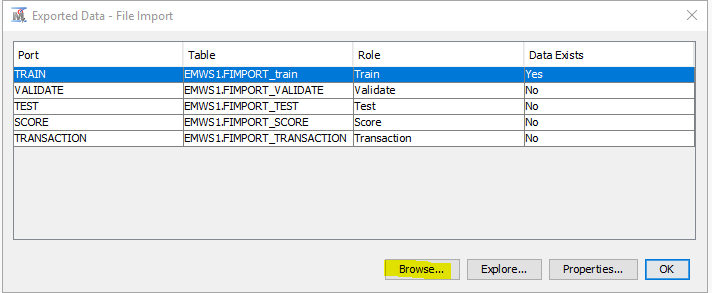
HW6: Document Classification

Use SAS Text Miner to cluster and classify documents while learning more about using SAS Text Miner. Complete the worksheet at the end of this assignment and enter your answers into the module assignment assessment.

30 points.

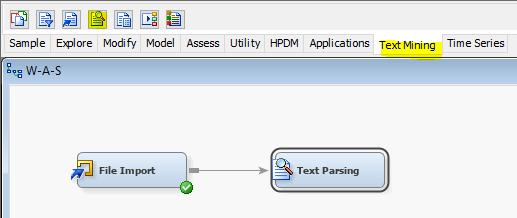
1. Launch SAS Enterprise Miner from <https://anywhereapps.niu.edu>
2. Create a new project 
3. Name your project (I called mine W-A-S, for weather, animal, sports). Select the SAS folder on your H drive as the SAS Server Directory. 
4. Create a new Diagram and drag a **File Import** node from the Sample tab of the SEMMA menu. [**VIDEO DEMONSTRATION OF STEPS 4-11**](https://youtu.be/AH-8zsubaq8)****
5. Select the Import File option from the File Import properties panel.
6. Browse to and select the **WeatherAnimalsSports** spreadsheet in your H:\SAS\SAS\_DATA folder. 
7. **Open** and **Ok**.
8. **Run** the File Import node.
9. To see the data set after the File Import node is run, go to the **Exported Data** line of the properties panel. Click the ellipsis button (…). Then select the **Train** data and click **Browse** near the bottom of the window. 
10. You see the rows of the data set. The first seven rows are shown below.

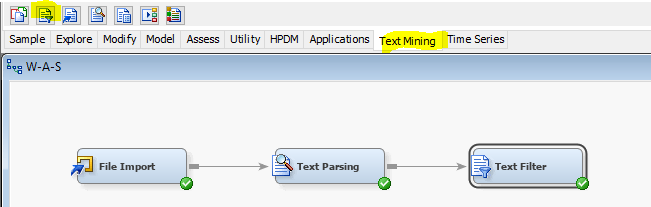
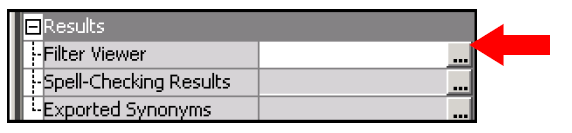
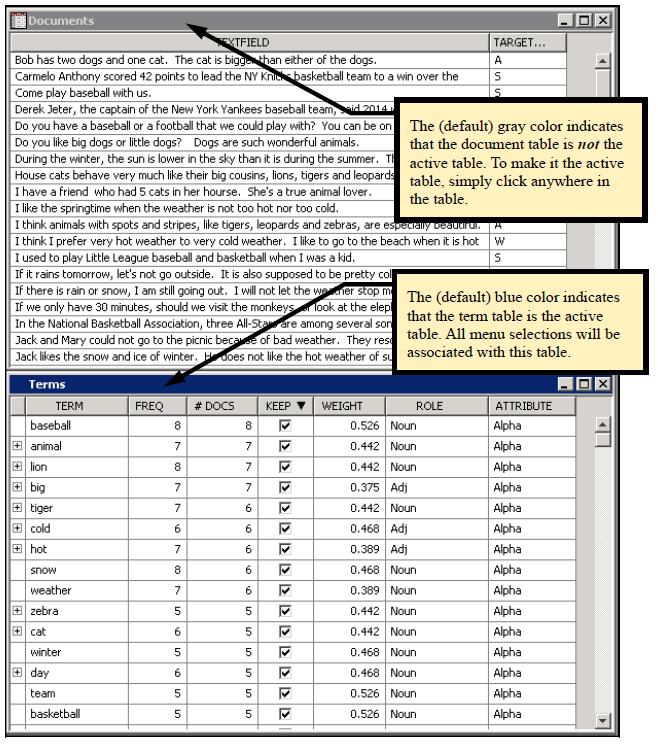
The data set has two fields: **Target\_Subject** (with values *A*, *S*, *W*) and **TextField**, which consists of short sentences. The sentences are about one of three subjects: animals(*A*), sports(*S*), and weather(*W*).

* It is important to understand that the **Target\_Subject** field was created by a person interpreting the content of each **TextField**. It was not created automatically by the Text Miner nodes. Consequently, the labeling of each document is subject to human error.



1. Read through a few of the rows and make sure that you understand the nature of the data set and how it is structured. The variable **TextField** is referred to as a *document*. All the rows of **TextField** together (47 rows of data) are referred to as the *corpus*.
2. Attach a **Text Parsing** node to the File Import node. This node has the language processing algorithms and has many different options that can be set by the user. For this demonstration, use the default settings. Run the Text Parsing node.



1. Attach a **Text Filter** node to the **Text Parsing** node. Change the **Minimum Number of Documents** value in the properties panel to **2**. This option filters out terms that are not used in at least two documents in the corpus collection. Because you use a very small data set, the default value 4 is too large and eliminates too many terms. Run the **Text Filter** node. 
2. Open **the Filter Viewer** in the properties panel. This is also called the *Interactive Filter Viewer*. 
3. Look at the two main windows that open in the Filter Viewer.You see what is shown in the display below. 
   * The (default) gray color indicates that the document table is ***not*** the active table. To make it the active table, simply click anywhere in the table. The (default) blue color indicates that the term table is the active table. All menu selections will be associated with this table. The first window, labeled **Documents**, simply lists each document and any other variables in the training data set(in this case, only the variable **Target\_Subject**). The second window, labeled **Terms**, gives information about each of the terms that came out of the Text Parsing node. A *term* does not have to be a single word. The term table contains the corpus dictionary—that is, it contains every term in the document collection defined by the training data set, after certain parts of speech have been eliminated.
   * The Terms window contains the following information:

**FREQ** = number of times the term appearsin the entire corpus.

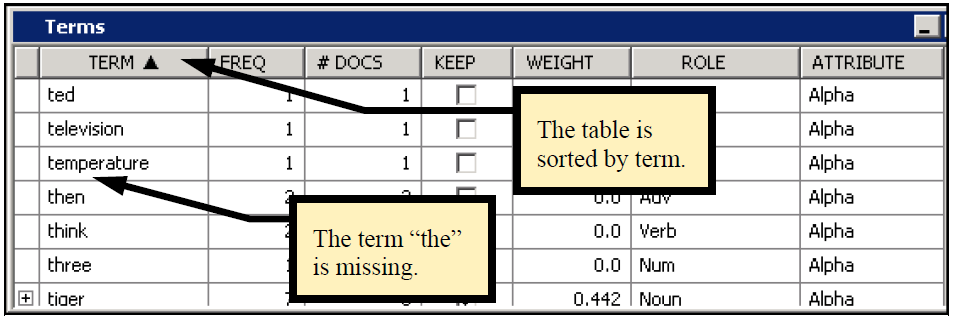
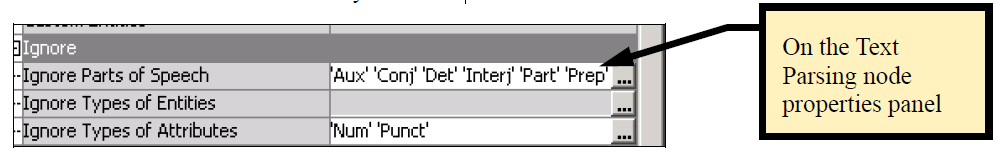
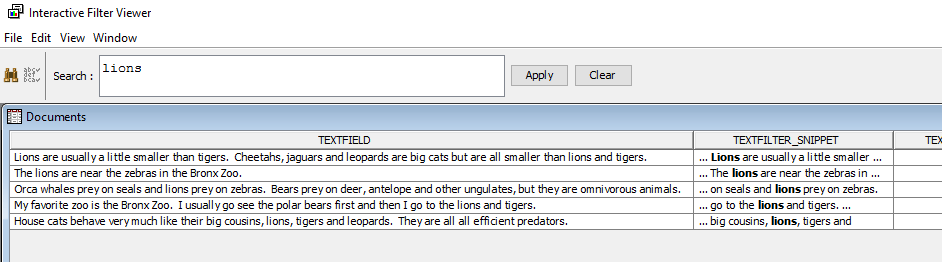
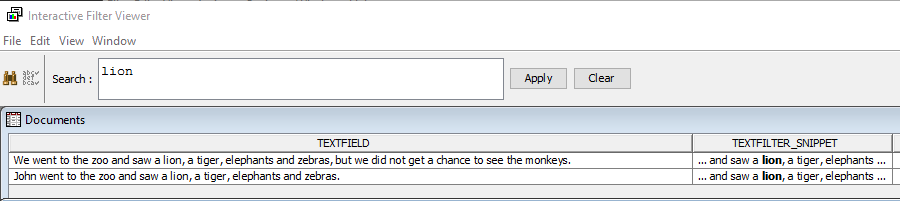
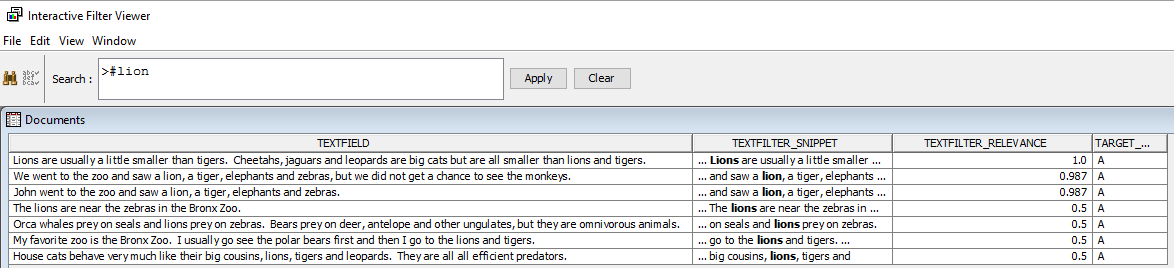
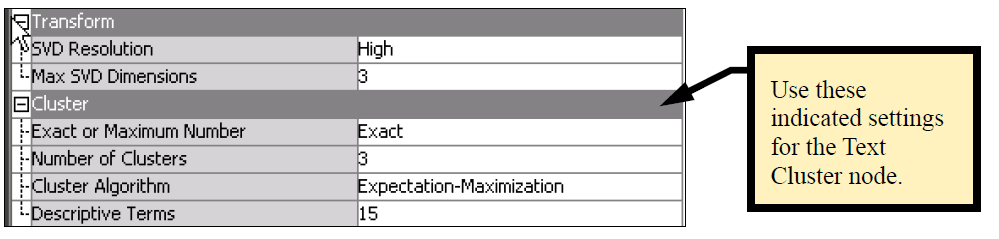
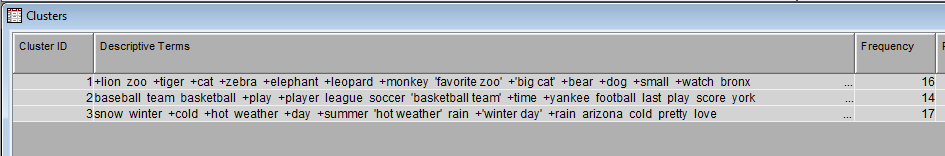
**#DOCS** = total number of documents in which the term appears.

**KEEP** = whether the term is kept for calculations. The keep status reflects whether a term is in the start list (KEEP=Y) or in the stop list (KEEP=N).

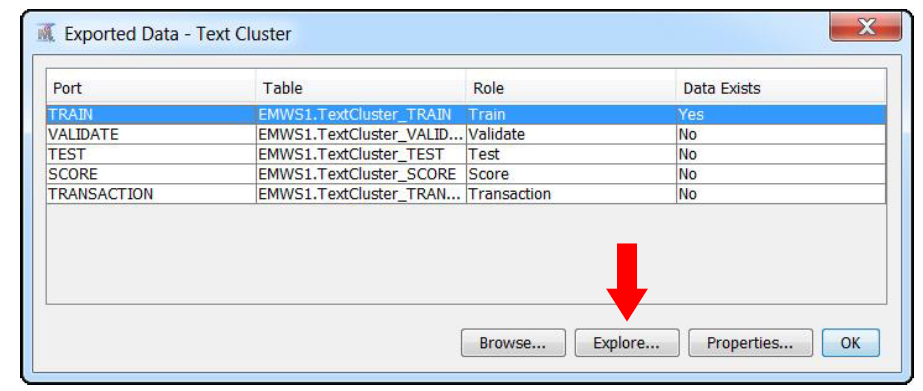
**WEIGHT** = a term weight. Term weights are explained later. The default term weight is mutual informationwhen a categorical target variable is present in the data.

**ROLE** = part of speech of the term.

**ATTRIBUTE**=one of abbr, alpha, mixed, num, or punct. Attributes num (number) and punct (punctuation) are ignored by default.

1. Go to the Terms window and confirm that the word “the “is not listed. (If the TERM column is not already in alphabetical order, you can sort a column by clicking on the heading.) 
2. Why does the most common word in the English language not appear in the list? To understand why, click the **Text Parsing** node so that the properties panel for that node is visible. Look at the properties near the bottom. You can see that there is an **Ignore Parts of Speech** property. By default, this excludes certain terms that are very common. 
   * In particular, 'Det' represents *Determiner*, which is a class of common words and phrases such as *the*, *that*, and *an*. These are eliminated unless you modify this property. Eliminating a word because of the **Ignore Parts of Speech** property is different from adding a word to the stop list. Words in the stop list appear in the term table, but are assigned a weight of zero. Words that are ignored are excluded from the term table. The distinction is that words in the stop list can be moved to the start list dynamically without re-parsing the document collection, whereas words excluded from the term table can be added only by modifying properties in the Text Parsing node and re-running the node. Because parsing typically consumes 80% to 90% of the processing time for a Text Miner process flow, you want to avoid re-parsing, especially for very large document collections.
3. Return to the Text Filter Viewer. In the query window, type **lions**. Click the **Apply** button. 
   * Five documents exhibit the word *lions*. The TEXTFILTER\_RELEVANCE score is a function of the word frequency normalized by the highest frequency encountered. For example, the word *lions* appears twice in the first document, and one time in the remaining documents. Thus, the last four documents have a relevance score of ½=0.5. The calculation is more complicated for compound queries.
4. Change the query to **lion**, and click **Apply**. 
   * Only two documents are returned. This verifies that the query is based on searching for words, not for sequences of characters. If the search was for any occurrence of the letters l-i-o-n, then seven documents would be found. The search feature applies what could be called a token-based search contrasted with a character-string-based search.
   * Queries are not case sensitive.
5. The operator ># can be used to find documents containing the word or any synonym or stemmed version of the word. Thus,**>#lion** returns all documents that contain *lion* or *lions* as words. 
6. **Clear** the search term, **Apply**, **Close** the Filter Viewer, **Save** if prompted.
7. The next few steps introduce the two main analytic text mining tools, the Text Clustering node and the Text Topic node. Attach a **Text Clustering** node to the **Text Filter** node. The Text Clustering node takes the 47 documents in the example data set and separates them into ***mutually exclusive*** and ***exhaustive*** groups (that is, clusters). The number of clusters to be used is under user control. Modify the four default settings. [VIDEO DEMONSTRATION OF STEPS 22-27](https://youtu.be/UijYPf4kjvE)
   1. Change **SVD Resolution** from **Low** to **High.**
   2. Change **Max SVD Dimensions** from **100** to **3.**
   3. Change **Exact or Maximum Number** (of clusters) to **Exact.**
   4. Change **Number of Clusters** from **40** to **3**
   * Regarding the Text Cluster properties, remember that you are using a very small and simple data set. You know that there are basically three types of documents (animals, sports, weather). It is reasonable to think in terms of creating a small number of clusters (for example, 3 to 5). Use **3**. In practice, with real and complex text data, you want to experiment with these parameters.
8. **Run** the **Text** **Cluster** node.
9. Open the Text Cluster node results when complete and examine the left side of the Clusters window as shown. 
   * Exactly three clusters were created, as requested in the properties panel. The **Descriptive Terms** column shows up to 15 terms that are given to help the user interpret the types of documents that are put into each cluster. (The number can be changed.) These terms are selected by the underlying algorithm as being the most important for characterizing the documents placed into a given cluster. Reading these, you can see that Cluster 1, which has 16 documents, has terms such as *favorite zoo*, *big cat*, and so on. These documents are likely about animals. The + indicates that a term has multiple versions either from stemming or from having synonyms. Cluster 2 has 14 documents that are likely related to sports. Cluster 3 has 17 documents that likely deal with weather.
10. To see the new variables that were generated by the Text Cluster node, close out of the results. Select **Exported Data** from the properties panel.

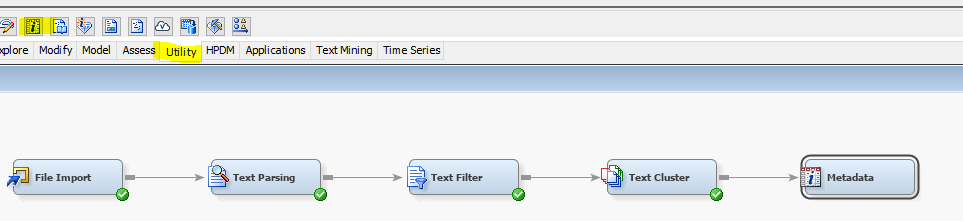


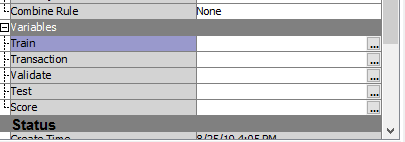
1. Select the **TRAIN** data set and click **Explore**. 
2. The upper right window (Sample Statistics) shows a list of variables that were exported from the Text Cluster node.

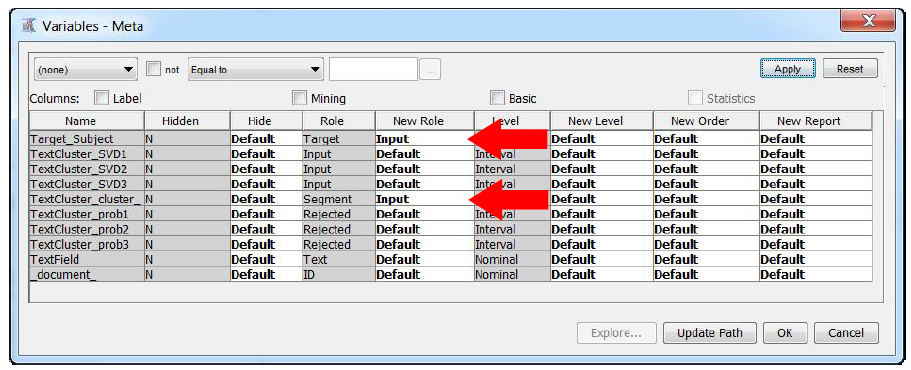


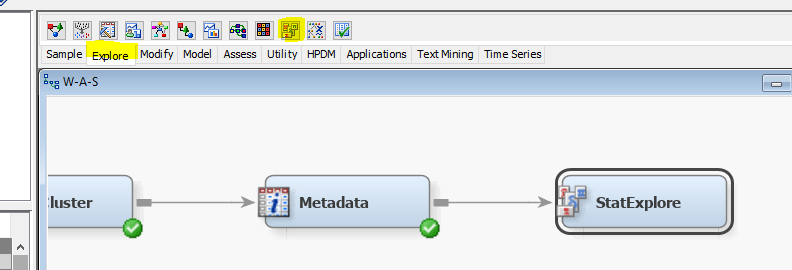
Several new variables have been added to the original variables **Target\_Subject** and **TextField**:

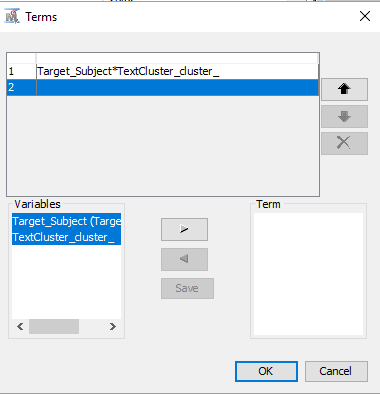
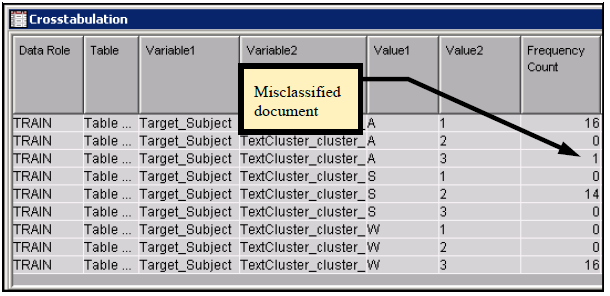
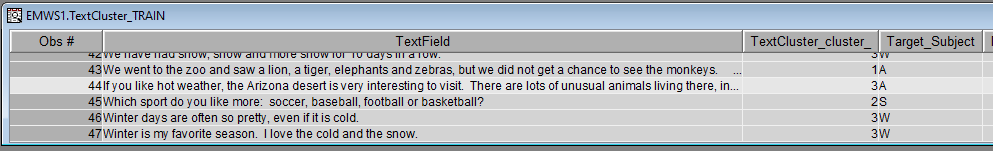
* + **TextCluster\_SVD1-TextCluster\_SVD3**–These are numeric variables calculated from a singular value decomposition of the (usually weighted) document-term frequency matrix. Each document is represented by its values on these three new variables. The values are also normalized so that for each document all the squared SVD values sum to 1.0. These are the variables that are used to cluster the documents. Because these SVD columns are added to the exported data, they can be used as input variables for any supervised or unsupervised learning node supported by Enterprise Miner.
  + **TextCluster\_cluster\_**–This is the Cluster ID, a categorical variable. In this example, it is simply a number from 1 to 3 because three clusters were created. The clusters were generated by performing a cluster analysis on the three **TextCluster\_SVD** variables. The interpretation of the clusters begins with looking at the descriptive terms given for each cluster, as you did earlier. The assigned numbers act as labels and are arbitrary. Any modification of the imported data, such as sorting differently, or adding a new document, can cause the labels to change, even if cluster membership does not change.
  + **TextCluster\_prob1-TextCluster\_prob3**–These variables are the probabilities of membership in each cluster for a given document. The sum of these probabilities is 1. A document is assigned to the cluster where it has the highest membership probability. These values are added when Expectation-Maximization clustering is used. Details of the two clustering algorithms supported by the Text Cluster node are provided later.
  + **\_document\_**–This is a document ID.

1. Close the results. On the Utility tab, drag a **Metadata** node into the diagram and attach it to the **Text Cluster** node. [VIDEO DEMONSTRATION OF STEPS 30-36](https://youtu.be/zi4rBF1-HhQ)
2. Select the ellipses (…) next to the Train field in the Metadata properties panel.



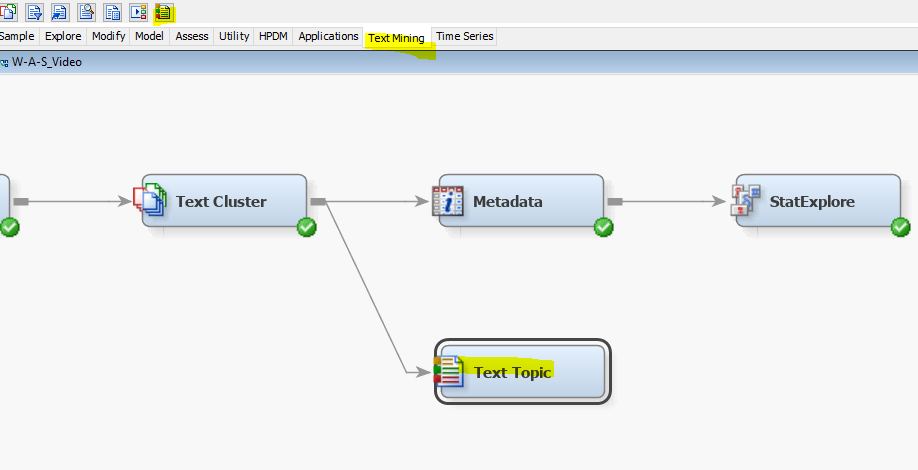
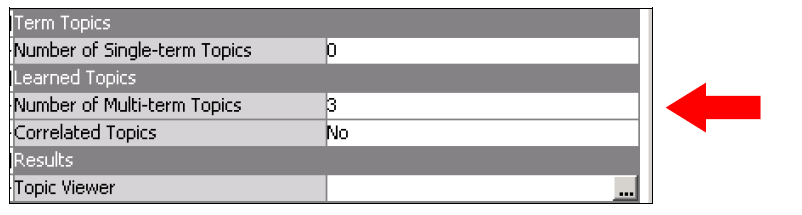
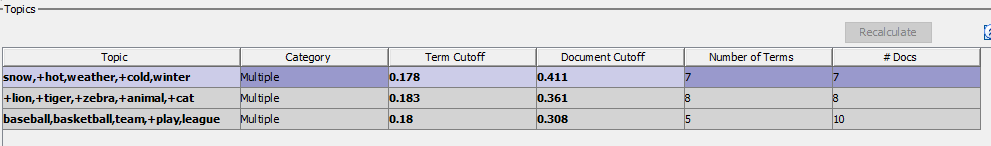
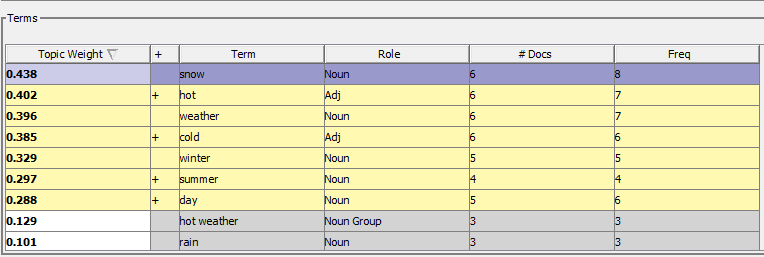
1. Change the role of **Target\_Subject** and **TextCluster\_cluster** to **Input**. 
2. Select **Ok** and **run** the **Metadata** node.
3. On the Explore tab, drag a **StatExplore** node into the diagram and attach

it to the **Metadata** node. 

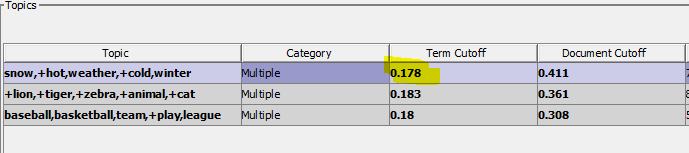
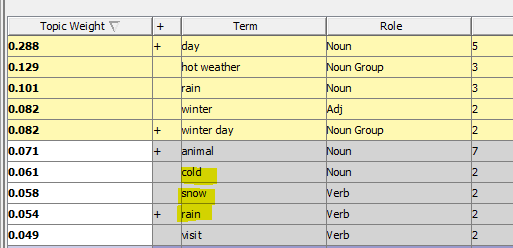
1. Select the **Cross Tabulation** property and add the **Target\_Subject\*TextCluster\_cluster\_** crosstabulation. To do so, select each input, select the right arrow, and then select **Save** when both inputs are selected. **Ok**. 
2. Run the **StatExplore** node and look at the results. 
   * This crosstabulation shows that cluster 1 (which was seen previously to have descriptive terms such as *favorite zoo*, *big cat*, and so on) consists of 16 documents defined that have to do with animals (*A*) as labeled by the human reader. Cluster 3(*hot weather*, *winter day*, and so on) consists of 17 documents, and 16 of them were defined as weather-related, and one cluster 1document was animal related. Cluster 2(*basketball team*, *play,* and so on) consists of 14 documents with a target value always equal to *S*. The three clusters line up almost perfectly with the labels given to the documents. There is a single misclassified document*.* ***It would be wonderful if real data worked out this well, but do not expect that!***
3. Because the data set is so small, you can simply examine the exported data to find the single misclassified document. In the **Text Cluster** properties panel, select **Exported Data**. Select the **TRAIN** data set and then click **Browse**. Scroll to the bottom. Notice that document 44 is the misclassified document (**\_TextCluster\_cluster\_=3, Target\_Subject=A**). 

44 in weather cluster (3), but was identified as an Animal document (A)

* + Document 44 mentions weather and animals, so it is an ambiguous document with respect to the three categories used. The human judge says that it is an animals document. The computer judge says that it is a weather document. It could be classified as both, so using mutually exclusive clustering is inadequate to properly score document 44.

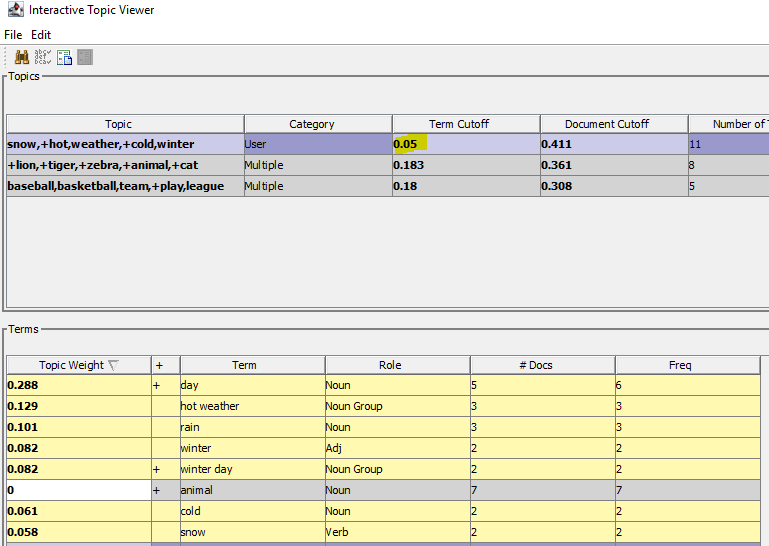
1. The Text Topic node is used to identify topics in a document collection. Although a cluster is a mutually exclusive category (that is, each document can belong to one and only one cluster), a document can have more than one topic or it can have none of the derived or user-specified topics. Attach a **Text Topic** node directly to the **Text Cluster** node. [VIDEO DEMONSTRATION OF STEPS 38-55](https://youtu.be/__YPwUjv3TQ) 
2. Make one change to the default properties by specifying **3** as the number of multi-term topics to create. Just as the number of clusters created is a parameter with which you want to experiment when you use the Text Cluster node, this parameter for the number of topics to create is typically something that you might try with different values. In this example, the artificial data set was purposely created with three different topics, so a reasonable value to start with would be 3 to 5 and not the default value of 25. You use 3. 
3. **Run** the **Text Topic** node.
4. After it completes, click the ellipsis for **Topic Viewer** on the properties panel. The Topic Viewer is an interactive group of three windows. The Topics window shows the topics created by the node. 
   * The three topics created by the algorithm also have key descriptive terms to guide interpretation. The five most descriptive terms for each topic are shown. By default, the first topic is selected when you open the viewer. In this example, the first topic has descriptive terms starting with *snow, hot, …,* and seems to relate to weather. The second topic has descriptive terms *lion*, *tiger, …*. This is evidently a topic related to animals. The descriptive terms for the third topic *(baseball, basketball*, …) are interpretable as having to do with sports. With this simple data set, the algorithm did very well in identifying what are known to be the three underlying topics in the documents. However, the **# Docs** column indicates that the node did a poor job of classifying documents. At most, 25 documents have been associated with one or more of the three topics, leaving at least 22 documents with no topic assignment.
   * In the Topics window, there is a column labeled **Term Cutoff**. For each created topic, the algorithm computes a topic weight for every term in the corpus. This measures how strongly the term represents or is associated with the given topic. Terms that have topic weights above the term cutoff appear in yellow in the Terms window shown below, and terms with topic weight below the cutoff appear in gray. 

Why aren’t “hot weather” and “rain” in the weather topic?

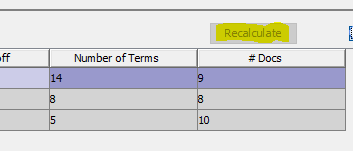
1. The term cutoff for the first topic is 0.178. You can manually change this value, and you are compelled to do so when you see that several terms below the cutoff seem to be associated with weather.
2. Manually change the **Term Cutoff** value for the first topic to **0.080**. Notice the new terms added to the weather topic when you press Enter. However, some obvious ones are still not included. 

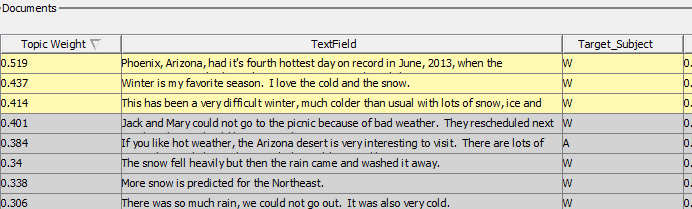
Change to 0.0

1. Another choice is to modify the topic weight for animal, perhaps changing it to zero because the word is not associated with weather. After “zeroing out” unrelated terms, you could then select a smaller cutoff. Set the **Topic Weight** of **animal** to **0.0**.
2. Now, change the **Term Cutoff** value to **0.05**. The following screen capture reflects the latter choice.

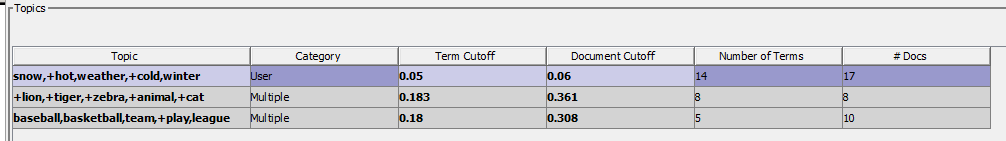


**animal** not included

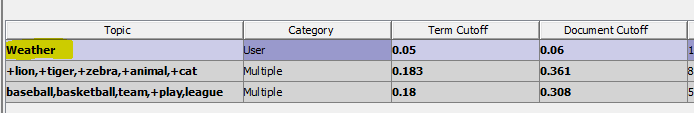
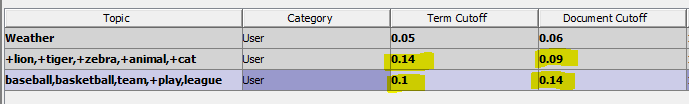
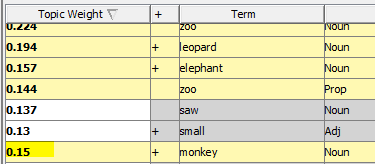
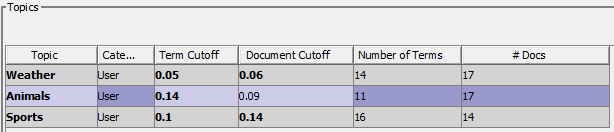
1. Press **Recalculate.** Notice the number # Docs value increased to 9, better classifying some previously missing documents. 
2. The manually entered changes add seven terms to the topic definition, but only increase the document count from seven to nine. You know that there are 16 weather documents, so additional changes are required to improve the topic definitions. The Document Cutoff for the first topic is 0.411. Examine

the document table. The 14 terms and their corresponding weights do a good job of rank ordering the documents with respect to weather. 

Many **W** documents not included

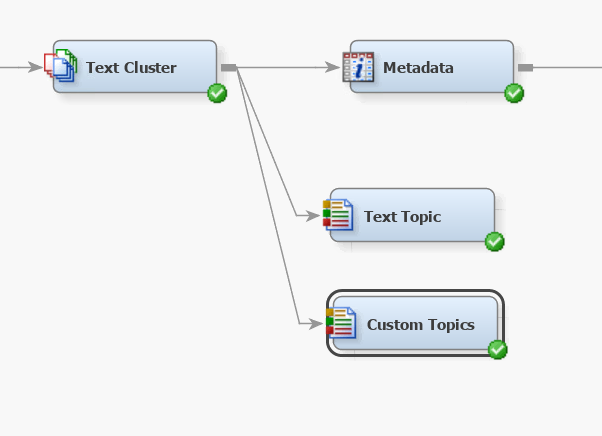
1. The last document selected to be associated with topic 1 has a topic weight of 0.414 and is shaded in yellow. Documents below the cutoff are shaded in gray. If you changed the Document Cutoff from 0.411 to 0.060, you would identify 17 documents as exhibiting topic 1, and all 17 would be in cluster 3 derived by the Text Cluster node, including the ambiguous document that has been labeled A. Change the **Document Cutoff** value to **0.060** and click **Recalculate**. 17 weather documents are now included.

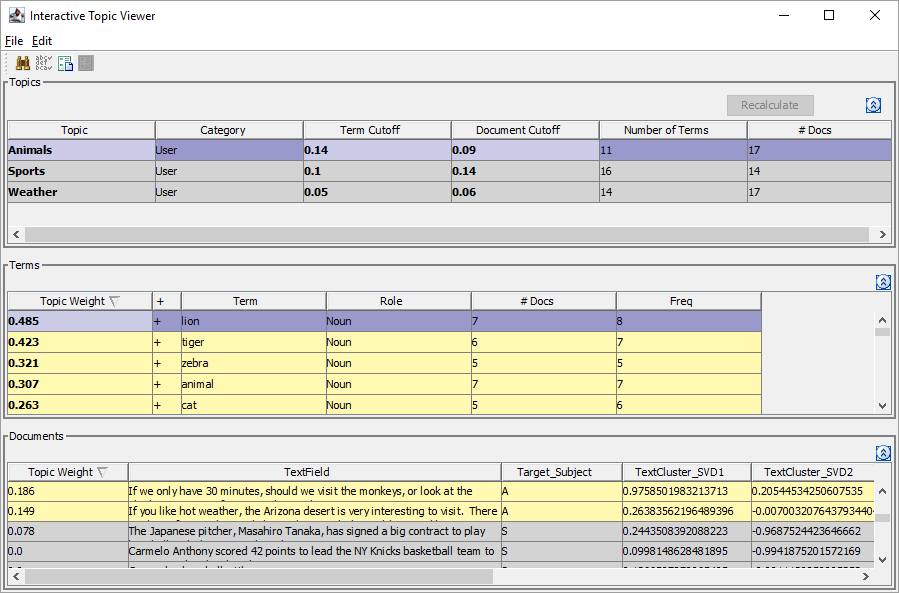
17 **W** Documents included!

1. Because you are using domain knowledge to exploit the successful rank ordering of documents into topic 1, the weather topic, you might as well edit the topic description. Click the cell for topic 1, and replace **snow,+hot,weather,+cold, winter** with **Weather**. 
2. The following topic table reflects changes to **Term Cutoff** and **Document Cutoff** for the remaining two topics. Change the **animals Term Cutoff** to **0.14**, and the **Document Cutoff** to **0.09**. Change the **sports Term Cutoff** to **0.1**, and the **Document Cutoff** to **0.14**. These changes improve the identification of documents related to the sports and animals topics. 
3. **Recalculate**. Notice the #Docs greatly improves.
4. You can make one more change to get seemingly perfect results. Select the animals topic. Change the **Topic Weight** for **monkey** to **0.150**. You change a Topic Weight by first double-clicking the Topic to activate it, then clicking the Topic Weight cell that you want to change, and then use the edit keys (Backspace and Delete) if necessary to type the replacement value. When you recalculate, you will see that 17 documents are classified as exhibiting the animals topic. There is one document that is flagged as both a weather topic and an animals topic. This ambiguous document is the same one that was identified by the StatExplore node. 
5. **Recalculate**.
6. Change the **Topic** names to reflect the **Animals** and **Sports** categories. 

17 **A** Documents included!

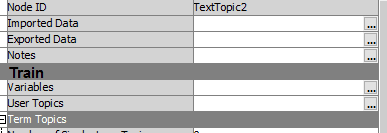
1. **Close** the **Topic Viewer** and **SAVE** the changes that you made.
2. To exploit the custom topic capabilities of the Text Topic Viewer, **copy and paste** the **Text Topic** node and attach the copied node to the Text Cluster node. **Rename** the new **Text Topic** node “**Custom Topics**.” Change the Number of **Multi-term Topics** to **zero**.



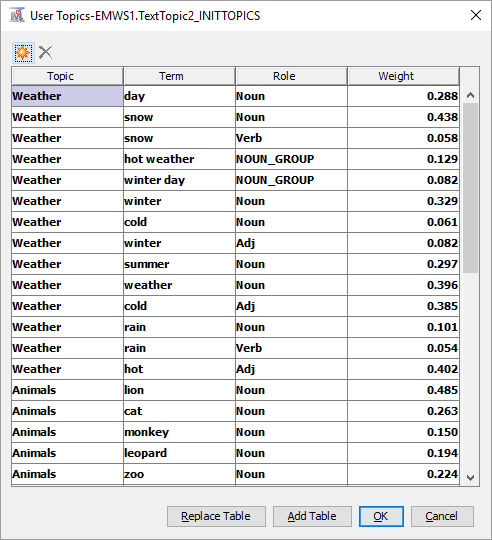
1. **Run** the new node, and then open the **Topic Viewer**.
2. The following table shows that the ambiguous document, the one beginning with “If you like hot weather,” is just above the Document Cutoff. 

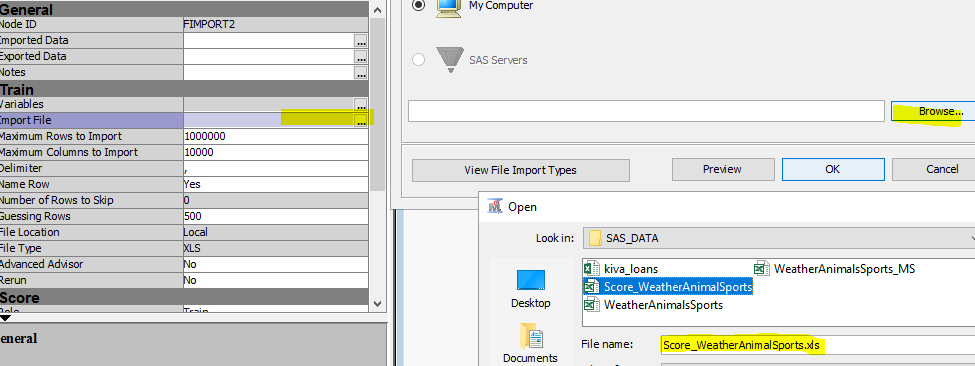
Just above Document Cutoff.

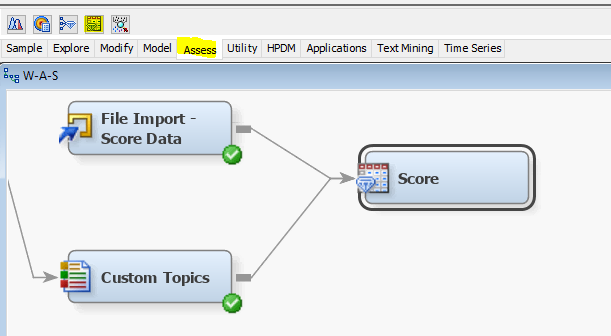
1. **Close** the **Topic Viewer**. Select **User Topics** from the properties panel.

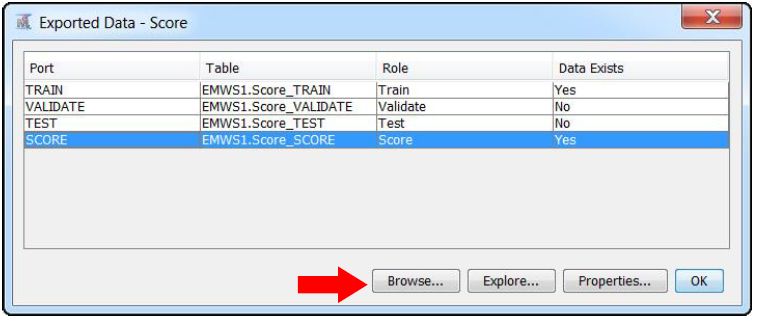


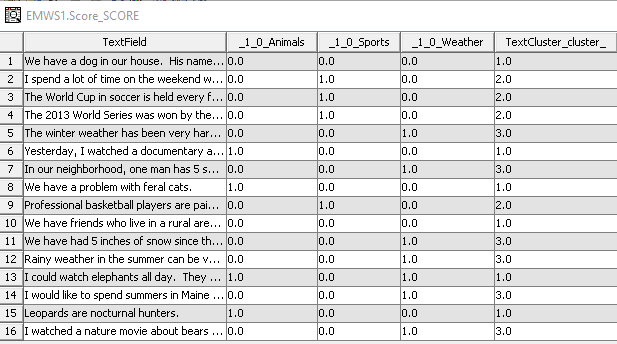
1. Observe the custom topic table created by your original edits in the Topic Viewer.



1. The final part of this demonstration is to use the Score node to score a new data set. Following the top part of the diagram shown at the beginning of this demonstration, bring in a new **File Import** node. **Rename** it “**File Import Score Data**”
2. Select **Import File** from the properties panel, select **Browse** and navigate to your **SAS/SAS\_DATA/** folder and select the **Score\_WeatherAnimalsSports** spredsheet. **Open**. 
3. In the properties panel, change the role of the data set to **Score.** 
4. **Run** the **File Import** node and look at the **Exported Data** window. Select **Train**, and **Browse**. This SCORE data set has 16 documents. They are related to the three subjects (animals, sports, or weather). (As is usually the case with a data set to be scored, there is no target field on this data set.)
5. The object now is to classify these documents using the previous analysis. To do that, bring in a **Score** node from the Assess tab and connect it to the output of **the File Import Score Data** node and also to the output of the **Custom Topics** Text Topic node.



1. **Run** the **Score** node. Then go to the **Exported Data** window through the properties panel. Select the **SCORE** data to view and click **Browse**. 
2. When the Browse window appears, move the column headings so that **TextField** is the first column heading and the other scored segment values are to the right of the text field. Recall that the clusters lined up as 1=animals, 2=sports, 3=weather. The custom topic segment variables are clearly labeled. Read through the 16 rows and check to see whether any of the classifications looks incorrect to you.



1. Do any of the topic indicators disagree with the cluster segments? Which observations appear to be misclassified? In particular, document 1 is classified as cluster 1 (animals), but the three binary text topic variables are zero, indicating that the first document exhibits none of the three topics. Document 7 is misclassified by the cluster ID and by the topic flags. Document 16 is ambiguous and could be both weather and animals. It is classified as a weather document by the cluster ID and the weather binary variable, but it could also be classified as an animals document.
2. Because of language challenges such as word sense disambiguation, the underlying text mining and modeling algorithms make mistakes. In this case, the very small number of training examples that were used likely influenced the quality of the results. Overall, the results look very promising!

Continue to the worksheet section below.

Worksheet Section

Apply the knowledge you gained from above to answer the following questions. Input your answers into the Module Assignment Worksheet assessment for this module.

1. A search for **LION** and **lion** return the same result? T/F \_\_\_True\_\_\_\_\_\_\_\_
2. A search for **Lion** and **lions** return the same result? T/F \_\_\_False\_\_\_\_\_\_\_\_
3. A search for **lion** and **>#lion** return the same result? T/F \_\_\_False\_\_\_\_\_\_\_
4. Documents mentioning sports were the highest frequency documents. T/F? \_\_\_\_\_\_\_False\_\_\_\_\_
5. If a topic has a Term Cutoff of .050, and there is a term that should be included in that topic, but its Topic Weight is .040, one way to include that term in that topic is to decrease the Term Cutoff of that topic to .039.

T/F? \_\_\_\_True\_\_\_\_\_

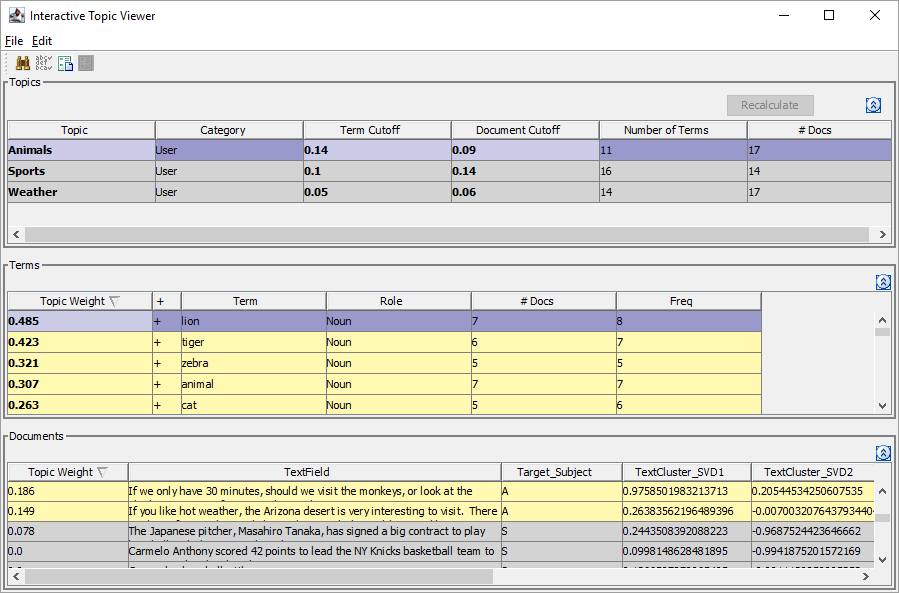


Figure 1: Image for reference

1. In the Score data, shown below, which topic has the most matched documents? ­­­­­­­­­­­\_\_\_\_\_\_\_Weather\_\_\_\_\_\_\_

