



1 Topic

“Text mining” with Knime and RapidMiner. Reuters Text Categorization.

The statistical approach of the "text mining" consists in to transform a collection of text documents in a matrix of numeric values on which we can apply machine learning algorithms¹.

The "unstructured document" designation is often used when one talks about text documents. This does not mean that he does not have a certain organization (titles, chapters, paragraphs, questions and answers, etc.). It shows first of all that we cannot express directly the collection in the form of a data table that is usually handled in data mining. To obtain this kind of data representation, a preprocessing phase is needed, then we extract relevant features to define the data table. These steps can influence heavily the relevance of the results.

In this tutorial, I take an exercise that I lead with my students for my text mining course at the University. We perform all the analysis under R with the dedicated packages for text mining such as “XML” or “tm”. The issue here is to perform exactly the study using other tools such as [Knime 2.9.1](#)² or [RapidMiner 5.3](#)³ (*Note: these are the versions available when I wrote the French version of this tutorial in April 2014*). We will see that these tools provide specialized libraries which enable to perform efficiently a statistical text mining process.

2 Dataset – Reuters collection

We use the well-known Reuters Text Categorization Dataset from the UCI repository⁴. The "reuters.xml" data file – in the XML format - is cleansed in order to simplify the processing.

It consists of 117 documents (novels). For each document is associated a topic ([SUJET](#) in French) and a newswire text ([TEXTE](#)). There are only two possible topics (categories): "acq" and "crude".

The aim of the text categorization - or more generally speaking “document classification” - is to build a predictive function which enables to predict the category of the document starting from the associated description (the newswire text in our context). Thus, we process a binary classification problem where [SUJET](#) is the class attribute.

Here are the two first observations of our collection:

¹ https://en.wikipedia.org/wiki/Text_mining

² <https://www.knime.org/> (Knime Analytcs Platform)

³ <https://rapidminer.com/> (RapidMiner Studio)

⁴ <https://archive.ics.uci.edu/ml/datasets/Reuters-21578+Text+Categorization+Collection>



```

<xml>
  <document>
    <sujet>acq</sujet>
    <texte>
      Resdel Industries Inc said
      it has agreed to acquire San/Bar Corp in a share-for-share
      exchange, after San/Bar distributes all shgares of its
      Break-Free Corp subsidiary to San/Bar shareholders on a
      share-for-share basis.
      The company said also before the merger, San/Bar would
      Barry K. Hallamore and Lloyd G. Hallamore, San/Bar's director
      of corporate development, 1,312,500 dlrs and 1,087,500 dlrs
      respectviely under agreements entered into in October 1983.
    </texte>
  </document>
  <document>
    <sujet>acq</sujet>
    <texte>
      Warburg, Pincus Capital Co L.P., an
      investment partnership, said it told representatives of Symbion
      Inc it would not increase the 3.50-dlr-per-share cash price it
      has offered for the company.
      In a filing with the Securities and Exchange Commission,
      Warburg Pincus said one of its top executives, Rodman Moorhead,
      who is also a Symbion director, met April 1 with Symbion's
      financial advisor, L.F. Rothschild, Unterberg, Towbin Inc.
      In a discussion of the offer, Warburg Pincus said Moorhead
      told the meeting there are no plans to raise the 3.50 dlr bid.
      Moorhead told the Rothschild officials that Warburg Pincus
      considers the offered price to be a fair one, Warburg Pincus
      said.
      Last Month Warburg Pincus launched a tender offer to buy up
      to 2.5 mln Symbion common shares.
    </texte>
  </document>

```

We note the hierarchical structure of the XML document:

```

<xml>
  <document>
    <sujet>
      ...
    </sujet>
    <texte>
      ...
    </texte>
  </document>
  ...
</xml>

```

The handling of the data file requires several steps:

1. We must parse and load it into a proper structure by separating the topic (SUJET) and text (TEXTE) for each document.



2. The subjects can be collected as it is in a vector, there is no particular difficulties here. It is therefore possible to perform simple statistic calculations. For instance, 71 (respectively 46) documents correspond to the subject "acq" (resp. "crude").
3. The texts can also be collected into vectors. But it is not possible to make statistical treatments at this step.
4. In order to achieve this, we must transform each text in a vector where the elements are labeled with "terms", common to all the documents, for which we assign values called "weights", specific to the handled document.
5. Last, we consolidate all the values in a data matrix where, for each line (document), we observe its class membership (topic) and the weights associated with the terms (columns). This is called "[document-term matrix](#)".

In the screenshot below, we show the first rows and columns in a table - generated under R using the "tm" package, the results will be a little different with Knime and RapidMiner according to the pre-treatments performed - with $n = 117$ lines (because there are 117 documents), $p = 2315$ terms ($p + 1 = 2316$ columns including the subject), with the weighting TF (term frequency) that corresponds to the number of occurrences of each term in the documents.

sujet	sanbar	corp	dlrs	hallamor	shareforshar	acquir	agre	agreement
acq	5	2	2	2	2	1	1	1
acq	0	0	0	0	0	0	0	0
acq	0	2	7	0	0	1	0	0
acq	0	1	0	0	0	0	0	0
acq	0	0	1	0	0	2	0	0

For example, the term "sanbar" appears 5 times in the first document, "corp" 2 times, "dlrs" 2 times, etc. Let us check this in the original document.

```
<document>
<sujet>acq</sujet>
<texte>
Resdel Industries Inc said
it has agreed to acquire San/Bar Corp in a share-for-share
exchange, after San/Bar distributes all shgares of its
Break-Free Corp subsidiary to San/Bar shareholders on a
share-for-share basis.
    The company said also before the merger, San/Bar would
Barry K. Hallamore and Lloyd G. Hallamore, San/Bar's director
of corporate development, 1,312,500 dlrs and 1,087,500 dlrs
respectviely under agreements entered into in October 1983.
</texte>
</document>
```



We observe that there is a rough correspondence between the term "sanbar", used in the data table, and the word "San/Bar" in plain text. This is a result of the cleanup performed prior to the generation of the data table (changing letters to lower case, stemming, lemmatization, removing stopwords, etc.). The goal is to reduce the number of terms (number of columns) of the matrix in order to obtain a more relevant description of the documents. The choices of the "terms" and the kind of "weighting" are crucial for the quality of the subsequent analysis.

3 Document classification with Knime

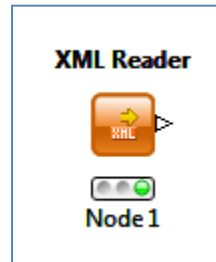
We use the [Knime](#) Analytics Platform, version 2.9.1 (downloaded in April 2014, when I wrote the French version of the tutorial). We detail the process leading to the generation of the documents-terms' matrix that will be used for the construction of a classifier allowing to predict the topics. Here are the main steps: reading and parsing the XML file, preparing documents in order to reduce the number of terms, extraction of terms, choice of the type of weighting and, finally, creation of the documents-terms matrix. Note: I have no doubt that there is a more direct way to lead these treatments. But because I must explain each operation, I adopt a step-by-step presentation so that everyone can trace and understand the whole process.

3.1 Importation of the XML file

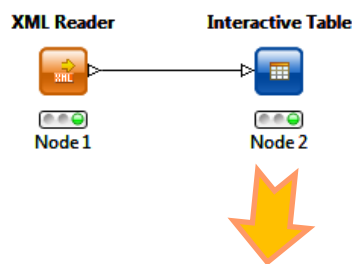
The screenshot displays the KNIME Analytics Platform interface. The main window shows a workflow with an XML Reader node. A context menu is open over the XML Reader node, with the 'Configure...' option selected. A dialog box titled 'Dialog - 2:1 - XML Reader' is open, showing the 'File' tab. The 'Selected File' field contains the path 't_for_soft_dev_and_comparison/text_mining/reuters.xml'. The 'Prefix of root's namespace' is set to 'dns'. The 'KNIME Console' at the bottom shows several warning messages: 'WARN File Reader The file 'file:/D:/DataMining/Databases_for_mining/benchmark_datasets/waveform/d...', 'WARN Logistic Regression Learner At least one column must be included.', 'WARN XML Reader No input file selected', and 'WARN XML Reader No input file selected'.



After we launch Knime, we create a new workflow (FILE / NEW). We set the name "Text Mining – Tutorial". We insert the XML / XML READER component into the workflow. We select the file "reuters.xml" into the dialog settings (contextual menu "Configure"). We click on the contextual menu "Execute". The indicator light becomes green if the operation is successful.



We use the component DATA VIEWS / INTERACTIVE TABLE to view the downloaded data. We click on the contextual menu "Execute and Open Views". All data are stored in a vector of size 1. The breakdown according to the documents is not done at this step.

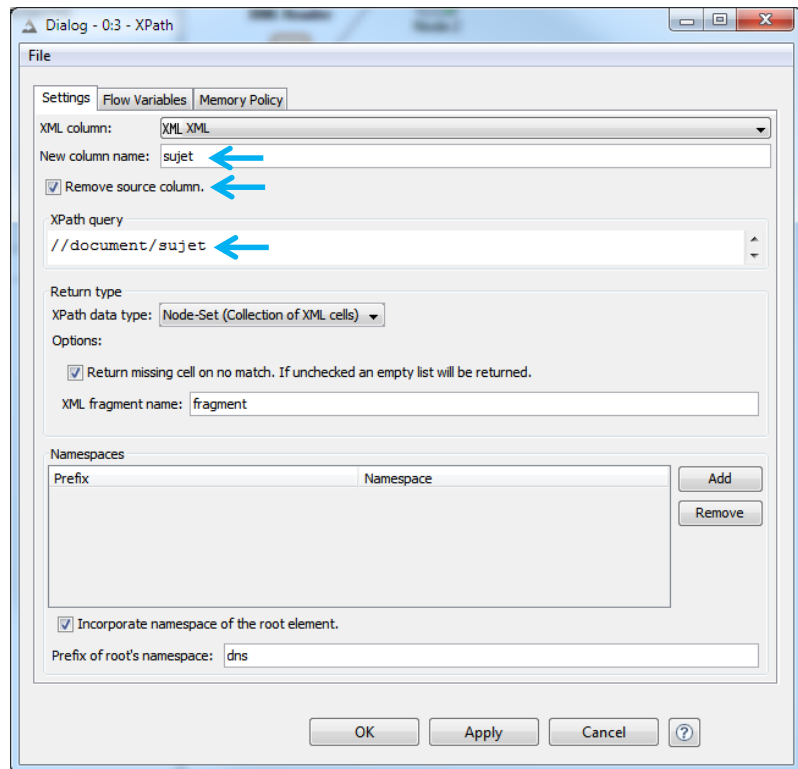
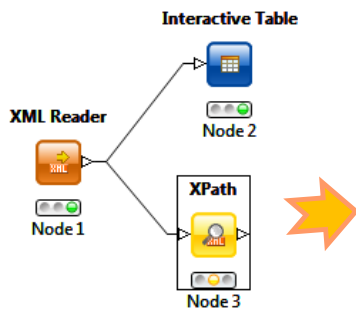


The screenshot shows a window titled "Table View - 2:2 - Interactive Table". The window has a menu bar with "File", "Hilite", "Navigation", "View", and "Output". The main area is a table with two columns: "Row ID" and "XML XML". The first row is labeled "Row0" and contains the following XML content:

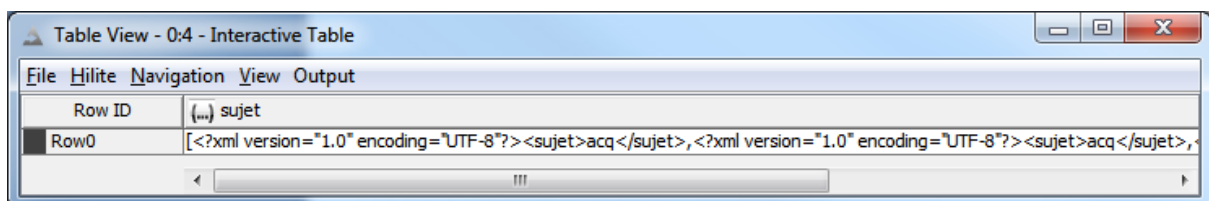
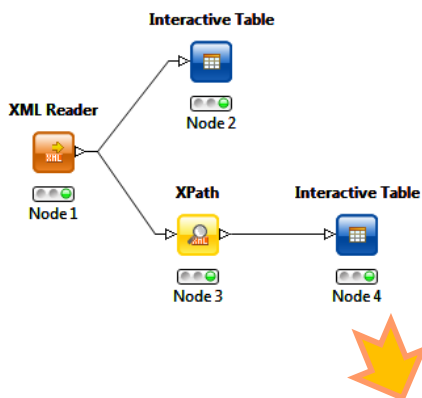
```
<?xml version="1.0" encoding="UTF-8"?>
<xml>
  <document>
    < sujet>acq</ sujet>
    < texte>
      Resdel Industries Inc said
      it has agreed to acquire San/Bar Corp in a share-for-share
      exchange, after San/Bar distributes all shgares of its
      Break-Free Corp subsidiary to San/Bar shareholders on a
      share-for-share basis.
      The company said also before the merger, San/Bar would
      Barry K. Hallamore and Lloyd G. Hallamore, San/Bar's director
```

3.2 Extracting the vector of subjects

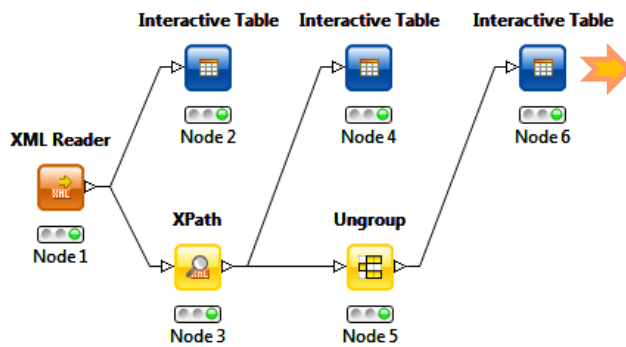
We use the tool XML / XPATH to extract the subjects associated with the documents. The settings (Configure contextual menu) are very important here: in "Xpath Query", we specify the field to retrieve; the vector is named SUJET (new column name); it replaces the data source (Remove Source Column).



The tool "Interactive Table" enables us to check the success of the processing. Only one row with all the subjects is available for the moment.

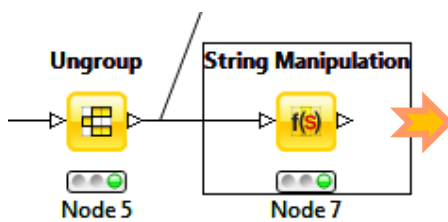


The component DATA MANIPULATION / ROW / TRANSFORM / UNGROUP (no particular settings) enables to split the subjects in a vector with 117 values.



Row ID	XML sujet
Row0_1	<?xml version="1.0" encoding="UTF-8"?>< sujet>acq</ sujet>
Row0_2	<?xml version="1.0" encoding="UTF-8"?>< sujet>acq</ sujet>
Row0_3	<?xml version="1.0" encoding="UTF-8"?>< sujet>acq</ sujet>
Row0_4	<?xml version="1.0" encoding="UTF-8"?>< sujet>acq</ sujet>
Row0_5	<?xml version="1.0" encoding="UTF-8"?>< sujet>acq</ sujet>
Row0_6	<?xml version="1.0" encoding="UTF-8"?>< sujet>acq</ sujet>
Row0_7	<?xml version="1.0" encoding="UTF-8"?>< sujet>acq</ sujet>
Row0_8	<?xml version="1.0" encoding="UTF-8"?>< sujet>acq</ sujet>
Row0_9	<?xml version="1.0" encoding="UTF-8"?>< sujet>acq</ sujet>
Row0_10	<?xml version="1.0" encoding="UTF-8"?>< sujet>acq</ sujet>
Row0_11	<?xml version="1.0" encoding="UTF-8"?>< sujet>acq</ sujet>
Row0_12	<?xml version="1.0" encoding="UTF-8"?>< sujet>acq</ sujet>

There are 117 lines in the table. We note however that the cells contain irrelevant characters for analysis. There is a cleaning to do. We proceed in two steps. First, we isolate the part after < sujet > of the string by using the DATA MANIPULATION / COLUMN / TRANSFORM / STRING MANIPULATION. We must set carefully the parameters here. The new column is called **SUJET2**.



String Manipulation Dialog - 0:7 - String Manipulation

Column List: ROWID, ROWINDEX, ROWCOUNT, XML sujet

Category: Extract

Function: substr(str, start), substr(str, start, length)

Description: Get length characters starting from start. start is zero based, i.e. to start from the beginning use start = 0. A negative value of

Expression: substr(\$sujet\$,46,20)

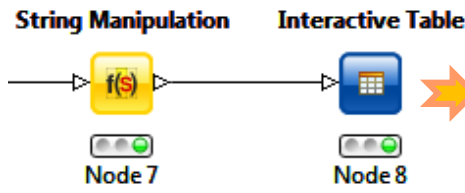
Flow Variable List: knime.workspace

Append Column: sujet2

Replace Column: XML sujet

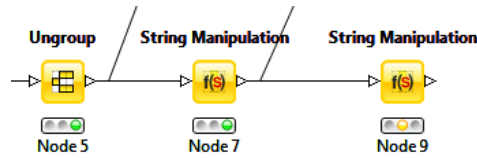
Buttons: OK, Apply, Cancel, ?

We can visualize the SUJET2 column with the INTERACTIVE TABLE tool.



Row ID	XML sujet	S sujet2
Row0_1	<?xml version="1.0" encoding="UTF-8"?>< sujet>acq</ sujet>	acq</sujet>
Row0_2	<?xml version="1.0" encoding="UTF-8"?>< sujet>acq</ sujet>	acq</sujet>
Row0_3	<?xml version="1.0" encoding="UTF-8"?>< sujet>acq</ sujet>	acq</sujet>
Row0_4	<?xml version="1.0" encoding="UTF-8"?>< sujet>acq</ sujet>	acq</sujet>
Row0_5	<?xml version="1.0" encoding="UTF-8"?>< sujet>acq</ sujet>	acq</sujet>
Row0_6	<?xml version="1.0" encoding="UTF-8"?>< sujet>acq</ sujet>	acq</sujet>
Row0_7	<?xml version="1.0" encoding="UTF-8"?>< sujet>acq</ sujet>	acq</sujet>
Row0_8	<?xml version="1.0" encoding="UTF-8"?>< sujet>acq</ sujet>	acq</sujet>
Row0_9	<?xml version="1.0" encoding="UTF-8"?>< sujet>acq</ sujet>	acq</sujet>
Row0_10	<?xml version="1.0" encoding="UTF-8"?>< sujet>acq</ sujet>	acq</sujet>
Row0_11	<?xml version="1.0" encoding="UTF-8"?>< sujet>acq</ sujet>	acq</sujet>
Row0_12	<?xml version="1.0" encoding="UTF-8"?>< sujet>acq</ sujet>	acq</sujet>

Second, we remove the characters after </sujet> by replacing them with an empty string using the STRING MANIPULATION tool.



Dialog - 0:9 - String Manipulation

String Manipulation | Flow Variables | Memory Policy

Column List: ROWID, ROWINDEX, ROWCOUNT, XML sujet, S sujet2

Flow Variable List: \$ knime.workspace

Category: Replace

Description: Replaces all occurrences of a String within another String.

Examples:

replace("abcabc", "ab", "cc") = "cc"

replace("abcabc", "ab", "zczc") = "zczc"

Function: replace(str, search, replace)

Expression: `replace($sujet2$, "</sujet>", "")`

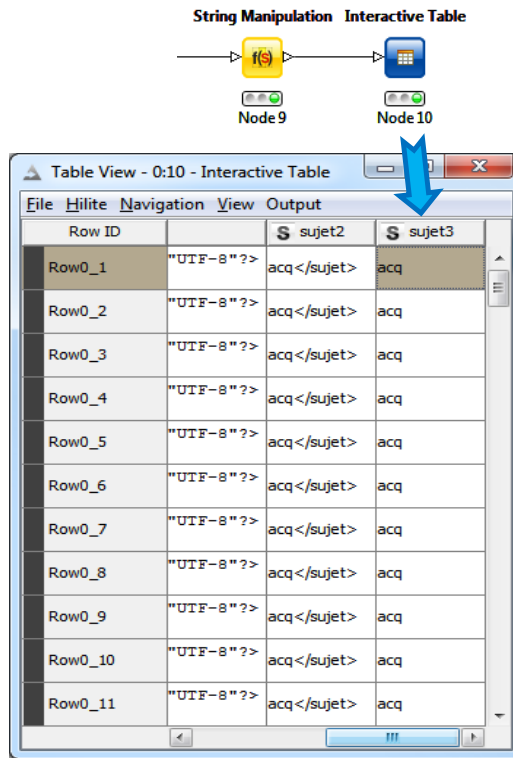
Append Column: sujet3

Replace Column: S sujet2

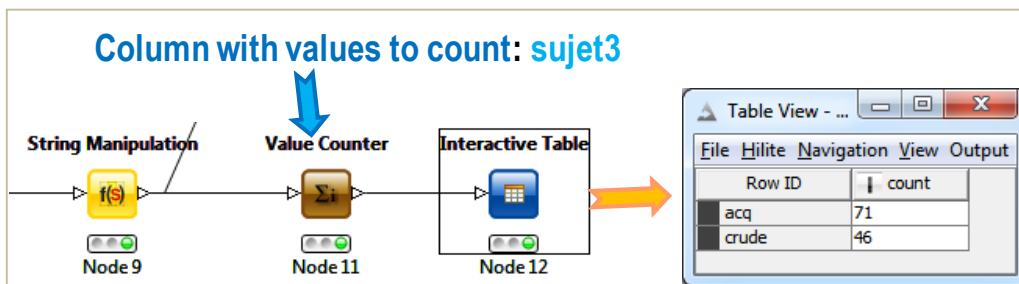
Buttons: OK, Apply, Cancel, ?



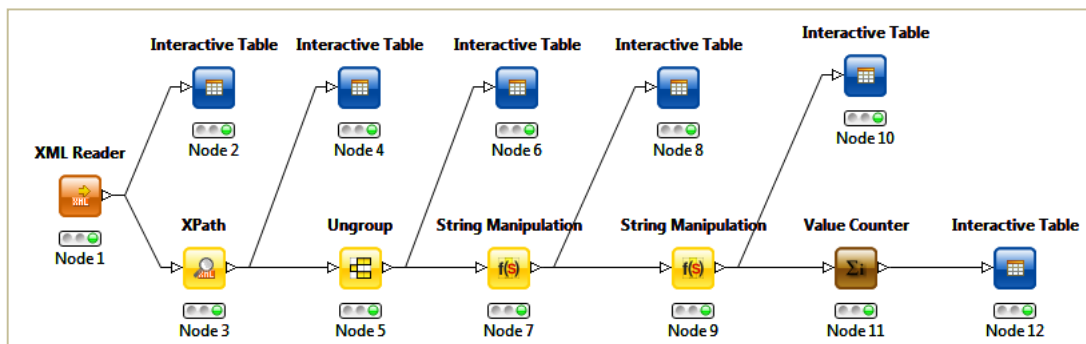
We visualize the **SUJET3** column.



We can perform a first statistical analysis. We calculate the frequency distribution of subjects by using the STATISTICS / COUNTER VALUE. We select the SUJET3 column.



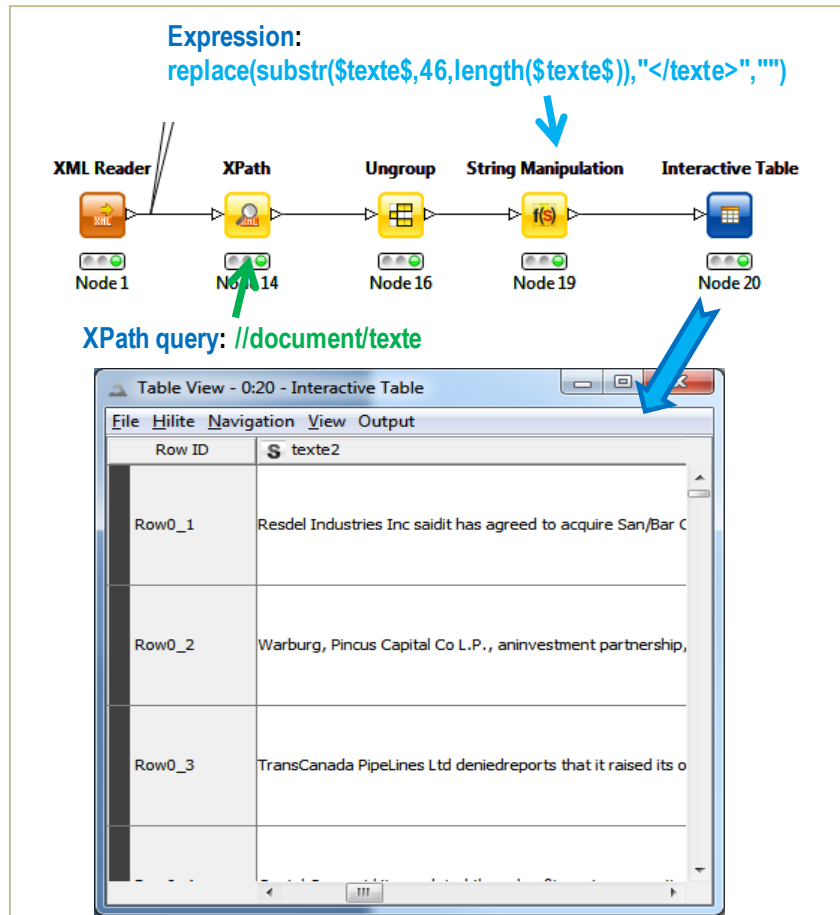
71 (resp. 46) newswires correspond to the topic "acq" (resp. "crude"). Many components were placed in the workspace. Here is the whole diagram at this stage of our analysis.



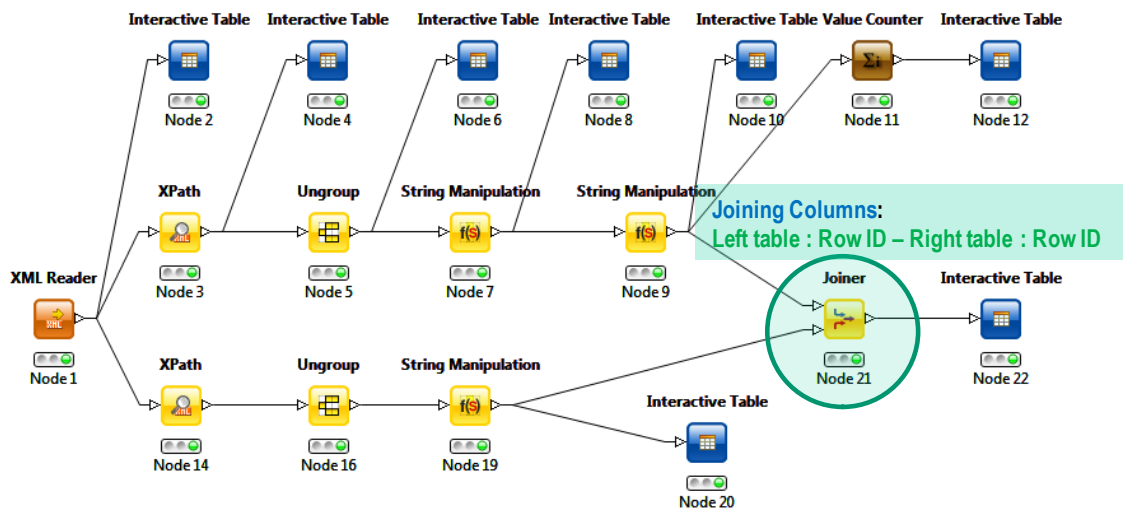


3.3 Extracting the vector of texts

Following the same approach, we will extract the part located between the <texte> and </texte> tags and store them in a second vector. We build the following sequence of nodes. The starting point is the "XML Reader" accessing the "reuters.xml" file.



3.4 Merging the two vectors in a unique table



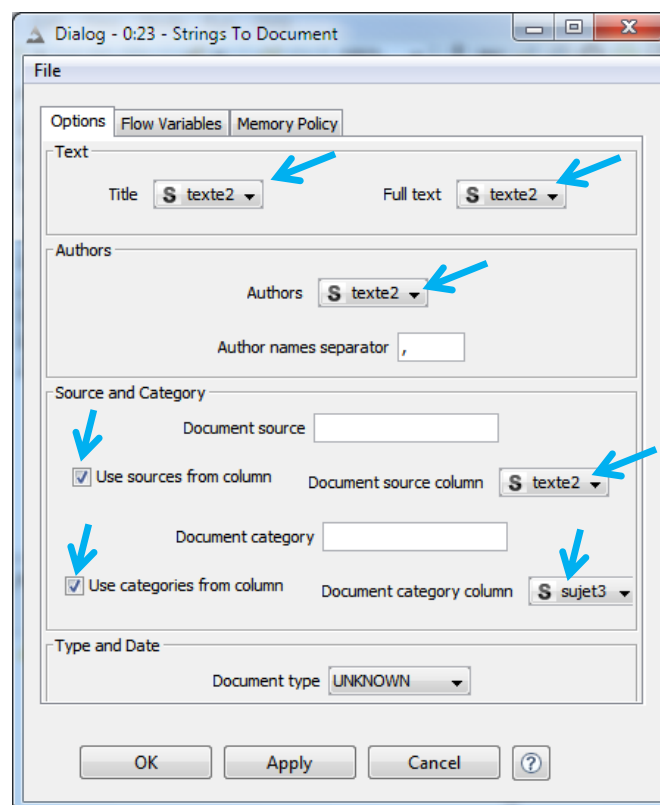


We want to perform a supervised learning process i.e. we want to use the “textes” to predict the “subjets”. Therefore, it is necessary to bring together the two vectors in a single data table making sure to match the lines (ROW ID). We use the tool DATA MANIPULATION / COLUMN / SPLIT & HANDSET / JOINER to do that.

3.5 Cleaning the texts

Before the creation of the document-term matrix, we must clean the texts. Several steps are needed:

- We must convert the text in an internal format “document” by specifying the different parts. We use the tool KNIME LABS / TEXT PROCESSING / TRANSFORMATION / STRINGS TO DOCUMENT. Here are the used settings. The designation of SUJET₃ as [document category column](#) is essential for the subsequent analysis.

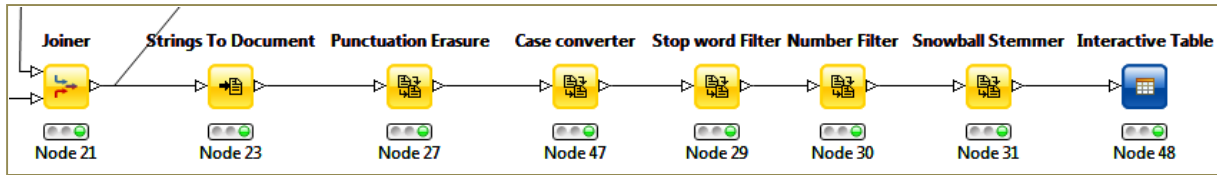


- Remove the punctuations by using the tool KNIME LABS / TEXT PROCESSING / PREPROCESSING / PUNCTUATION ERASURE.
- Change the letters to lower case: DATA MANIPULATION / COLUMN / TRANSFORM / CASE CONVERTER.
- Remove stop words with KNIME LABS / TEXT PROCESSING / PREPROCESSING / STOP WORD FILTER. We use the internal list of English words (Use build in list – Stop word lists: English).
- Remove numbers: KNIME LABS / TEXT PROCESSING / PREPROCESSING / NUMBER FILTER.



- Stemming words: KNIME LABS / TEXT PROCESSING / PREPROCESSING / SNOWBALL STEMMER (<https://en.wikipedia.org/wiki/Stemming>).

Here is the workflow starting from the JOINER node.



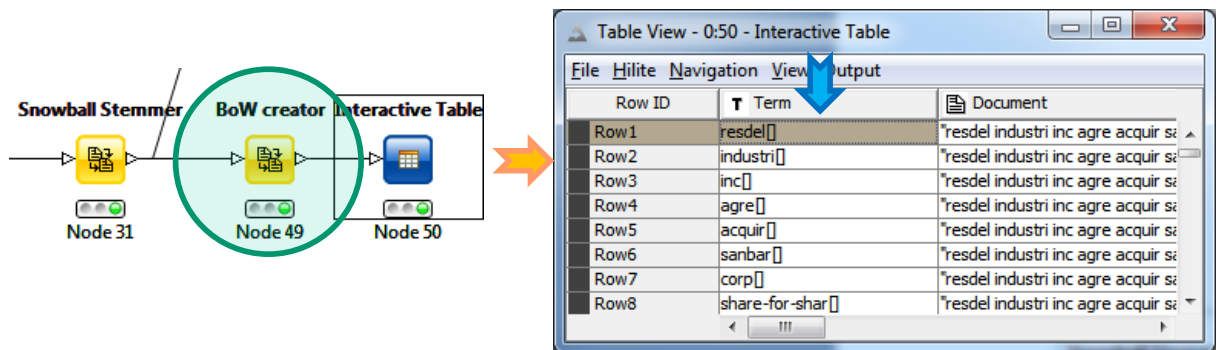
The cleaning is particularly radical. We barely acknowledge the text if we compare the first newswire with the original form. But this processing is supposed to allow distinguish the “useful” information from the “noise”. We expect that the relevant information for statistical analysis will be available after these pretreatments.

Resdel Industries Inc said it has agreed to acquire San/Bar Corp in a share-for-share exchange, after San/Bar distributes all shgares of its Break-Free Corp subsidiary to San/Bar shareholders on a share-for-share basis. The company said also before the merger, San/Bar would Barry K. Hallamore and Lloyd G. Hallamore, San/Bar's director of corporate development, 1,312,500 dlrs and 1,087,500 dlrs respectviely under agreements entered into in October 1983.

Row ID	Document
Row0_1	*resdel industri inc agre acquir sanbar corp share-for-sharexchangsanbar distribut shgare t
Row0_2	*warburgpincus capit co lpinvest partnershipold repres symbioninc increas 350-dlr-per-sha
Row0_3	*transcanada pipelin ltd denireport rais offer dome petroleum ltd ltdmpgtbillion canadian dlrs
Row0_4	*centel corp complet sale water properti serv custom southwestern kansa communiti centra
Row0_5	*pbs build system america incanaheimcalifcompanitold secur exchangcommiss acquir share r
Row0_6	*csr ltd ltcsrasgtintend proceed plan bid build materi monier ltdltnmrasgtdespit counter-bid l
Row0_7	*ltvirginia feder save loanassociationtsgn definit agreement acquir ltmontros hold cogtaffi
Row0_8	*allied-sign inc agre sell amphenol product unit subsidiari lpi invest ltlpigtwallingfordconnive
Row0_9	*lloyd invest manag ltdlondon-bas invest firmrais stake italfund sharepct total outstandcom
Row0_10	*arthur appletonchicago investortold secur exchang commiss acquirshare sage drill co incpc
Row0_11	*commonwealth aluminumcomalcolgoldendalwashsmelter market would-b buyerolumbia alur

3.6 Extracting the terms

We use the tool KNIME LABS / TEXT PROCESSING / TRANSFORMATION / BOW CREATOR in order to extract the terms (words) which appear at least one times in all of the documents.



We get a table with the list of terms and documents where they appear.

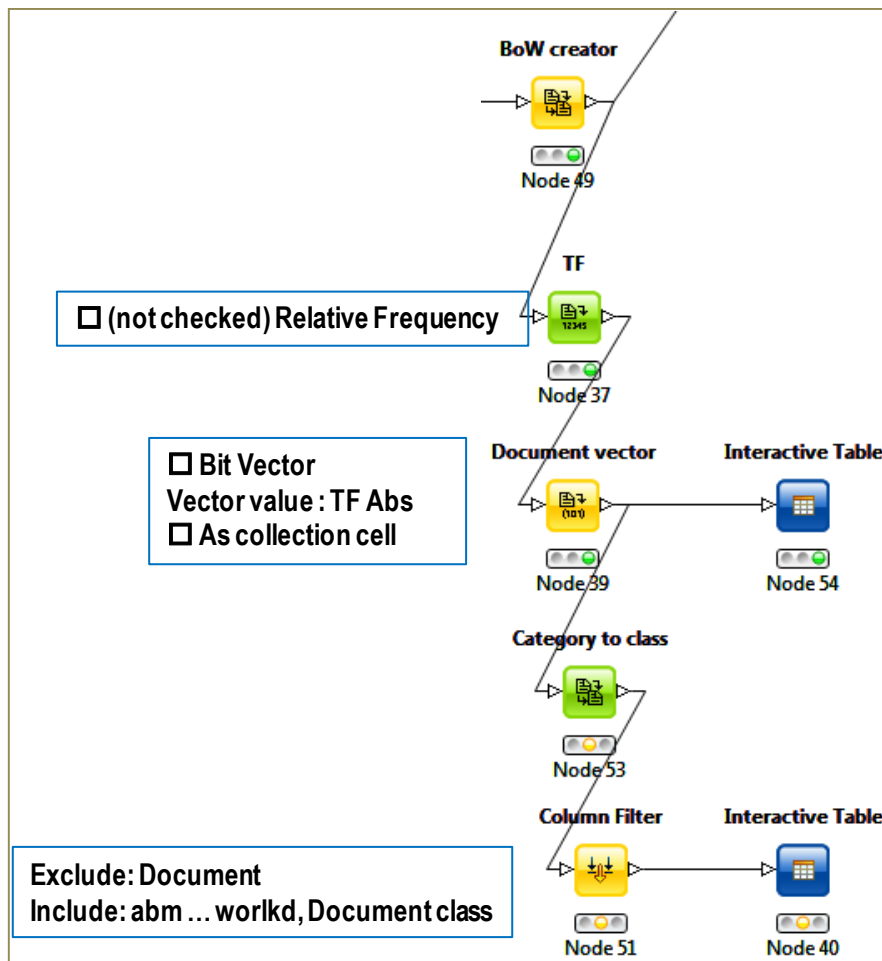


3.7 Term frequency weighting – Creation of the document term matrix

We want to create the document term matrix. We choose the TF (term frequency) weighing first i.e. we count the number of times each term occurs in each document. We use the following nodes:

- KNIME LABS / TEXT PROCESSING / FREQUENCIES / TF enables to count the number of occurrence of the terms (TF ABS column).
- KNIME LABS / TEXT PROCESSING / TRANSFORMATION / DOCUMENT VECTOR creates the DT matrix. We set as VECTOR VALUE the output of the preceding node (TF ABS).
- KNIME LABS / TEXT PROCESSING / MISC / CATEGORY TO CLASS enables to transform the category assigned to the documents (see STRINGS TO DOCUMENT in section **Erreur ! Source d u renvoi introuvable.**) into a class attribute i.e. the target variable for the supervised learning algorithm.
- Last, DATA MANIPULATION / COLUMN / FILTER / COLUMN FILTER enables to exclude the “document” column which is unused for the remainder of the analysis.

Here is the sequence of nodes. We highlight the parameters modified compared with the default values for each tool.





With the INTERACTIVE TABLE node, we visualize the document term matrix. The class membership of the documents appears into the last column.

Row ID	alid	D highlight	D play	D workkd	S Docum...
1		0	0	0	acq
2		0	0	0	acq
3		0	0	0	acq
4		0	0	0	acq
5		0	0	0	crude
6		0	0	0	acq
7		0	0	0	crude
8		0	0	0	acq
9		0	0	0	acq
10		0	0	0	acq
11		0	0	0	crude

We have $n = 117$ instances and 2418 terms/descriptors (+ the target variable).

3.8 Modeling with decision trees

We intend to use the J48 algorithm of the WEKA extension (that we must install before). "Document class" is the target attribute.

The workflow diagram on the left shows a 'Column Filter' node (Node 51) connected to an 'Interactive Table' node (Node 40) and a 'J48 (3.7)' node (Node 55). The dialog box on the right, titled 'Dialog - 0:55 - J48 (3.7)', shows the configuration for the J48 algorithm. The 'Select target column' dropdown is set to '\$ Document class', indicated by a blue arrow.

We obtain the flowing decision tree.



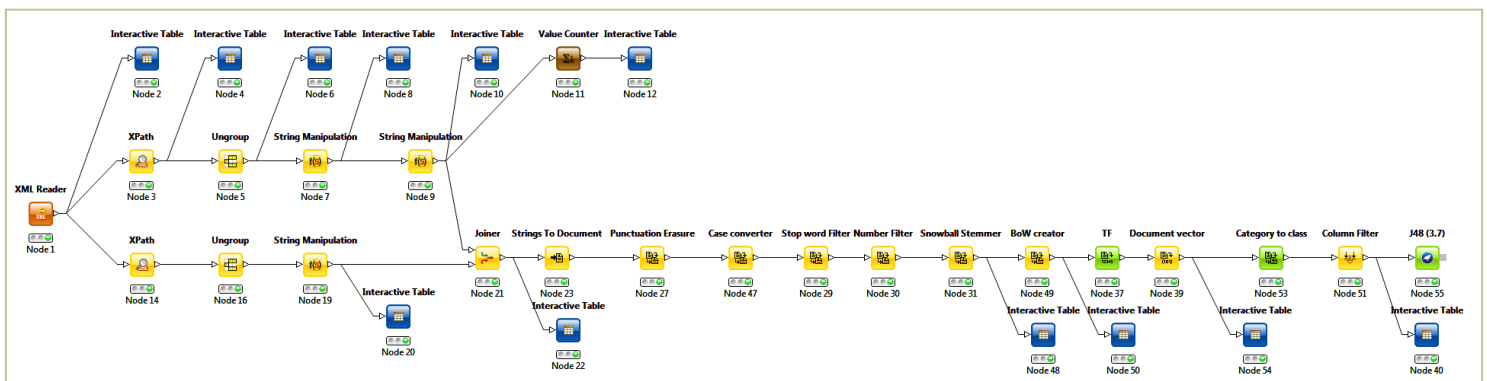
```
Weka Node View - 0:55 - J48 (3.7)
File
Weka Output | Graph | Summary | Source | Additional Measures
J48 pruned tree
-----
oil <= 0: acq (52.0/1.0)
oil > 0
|   plc <= 0
|   |   pacif <= 0
|   |   |   cooper <= 0
|   |   |   |   buy <= 0
|   |   |   |   |   cash <= 0
|   |   |   |   |   |   agre <= 0: crude (43.0/1.0)
|   |   |   |   |   |   |   agre > 0: acq (3.0/1.0)
|   |   |   |   |   |   |   |   cash > 0: acq (4.0/1.0)
|   |   |   |   |   |   |   |   |   buy > 0: acq (3.0/1.0)
|   |   |   |   |   |   |   |   |   |   cooper > 0: acq (3.0)
|   |   |   |   |   |   |   |   |   |   |   pacif > 0: acq (3.0)
|   |   |   |   |   |   |   |   |   |   |   |   plc > 0: acq (6.0)
Number of Leaves :      8
Size of the tree :     15
```

We find that: (1) The document is assigned to the subject “**crude**”: **IF** the term “oil” appears (at least one times) **AND** (plc, pacif, cooper, buy, cash, agre) does not appears into the document. (2) In all other cases, the document is assigned to “**acq**”.

These decision rules are easy to understand and to deploy.

3.9 First assessment

Let us recap the treatments performed to achieve this result. The INTERACTIVE TABLE components are used to visualize and check the operations at each step⁵.



As always, the sequence seems very simple after the fact. The most difficult finally under KNIME has been to identify the right tool for each treatment and to specify the appropriate settings. The need to categorize documents (section **Erreur ! Source du renvoi introuvable.**) before to generate the t

⁵ The whole diagram in the Knime format is available online.

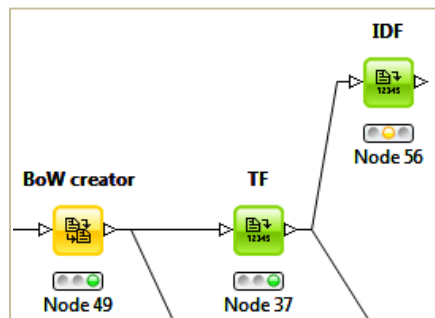


arget variable has not been easy to find. I do not know if there is another simpler way to create the target variable.

3.10 TF-IDF weighting

Now, we want to experiment the [TF-IDF](#) weighting scheme.

Calculation of the IDF (Inverse Document Frequency). We use the tool KNIME LABS / TEXT PROCESSING / FREQUENCIES / IDF to calculate the inverse document frequency. We set the tool after the TF node, we will combine them later. There is no specific parameter to specify.



Calculation of the weighting TF-IDF. We associate the two weighting scheme by using the following formula (*other formulas are possible, see <http://en.wikipedia.org/wiki/Tf-idf>*):

$$\text{TFIDF} = \text{LOG}_{10}(1 + \text{TF}) * \text{IDF}$$

Dialog - 0:59 - Java Snippet

File

Java Snippet Additional Libraries Templates Flow Variables Memory Policy

Create Template...

Column List

- ROWID
- ROWINDEX
- ROWCOUNT
- T Term
- Document
- TF abs
- IDF

Flow Variable List

- knime.workspace

```

1 // system imports
12 // Your custom imports:
13
14 // system variables
26 // Your custom variables:
27
28 // expression start
30 // Enter your code here:
31 out_prod = Math.log10(1+c_TFabs)*c_IDF;
32
33
34
35
36 // expression end

```

Input

Column / Flow variable	Java Type	Java Field	Add	Remove
TF abs	Integer	c_TFabs		
IDF	Double	c_IDF		

Output

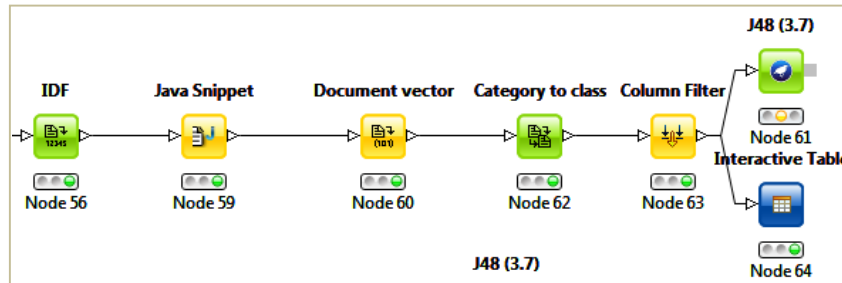
Field ...	Column / Flo...	Output Type	Java Type	Java Field	Add	Remove
<input type="checkbox"/>	tfidf	<input checked="" type="checkbox"/> DoubleCell	<input type="checkbox"/> Double	out_prod		

OK Apply Cancel ?



We use the node MISC / JAVA SNIPPET / JAVA SNIPPET in order to create the new variable. This tool seems very powerful and enables to perform a wide variety of operations.

The rest of the workflow. As previously, we perform the remainder of the operations by using the nodes: DOCUMENT VECTOR (**Vector value = TFIDF**; attention to the settings), CATEGORY TO CLASS, COLUMN FILTER and J48.



Here is a partial view of the document-term matrix. The weights are of course different from the TF weighting...

Row ID	D abm	D gold	D corp	D proceed	D initi	D public	D offer	D seven	D mln
118	1.751	1.213	0.495	0.873	0.706	0.596	0.66	0.764	0.304
119	0	0	0	0	0	0	0.451	0	0.415
120	0	0	0.41	0.596	0	0.596	0	0	0.535
121	0	0	0	0	0	1.055	0.798	0	0.608
122	0	0	0.28	0	0	0	0	0	0.415
123	0	0	0.28	0	0	0	0.451	0	0
124	0	0	0	0	0	0	0	0	0.304
125	0	0	0	0	0	0	0	0	0
126	0	0.872	0.559	0	0	0	0	0	0.304
127	0	0	0	0	0	0	0	0	0.484
128	0	0	0	0	0	0	0	0	0
129	0	0	0.28	0	0	0	0	0	0.304

...but the decision tree is the same because the splitting rules for the nodes are based only on the presence or absence of the terms (TF > 0 or not).

```

Weka Node View - 0:61 - J48 (3.7)
File
Weka Output Graph Summary Source Additional Measures

J48 pruned tree
-----

oil <= 0: acq (52.0/1.0)
oil > 0
|
|   plc <= 0
|   |
|   |   pacif <= 0
|   |   |
|   |   |   cooper <= 0
|   |   |   |
|   |   |   |   buy <= 0
|   |   |   |   |
|   |   |   |   |   cash <= 0
|   |   |   |   |   |
|   |   |   |   |   |   agre <= 0: crude (43.0/1.0)
|   |   |   |   |   |   |
|   |   |   |   |   |   |   agre > 0: acq (3.0/1.0)
|   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   cash > 0: acq (4.0/1.0)
|   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   buy > 0: acq (3.0/1.0)
|   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |   cooper > 0: acq (3.0)
|   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |   |   pacif > 0: acq (3.0)
|   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |   |   |   plc > 0: acq (6.0)

Number of Leaves :      8
Size of the tree :     15

```



Since we have a more elaborate weighting scheme. Let us see what happens when we use a SVM (support vector machine) linear for example. We use the WEKA component / WEKA (3.7) / CLASSIFICATION ALGORITHMS / FUNCTIONS / SMO (3.7). We specify a linear kernel and we do not normalize the data.

The diagram illustrates a Weka workflow. It starts with a 'Column Filter' (Node 63) which branches into two paths. One path goes to 'SMO (3.7)' (Node 65), and the other goes to 'J48 (3.7)' (Node 61). The 'SMO (3.7)' node also feeds into an 'Interactive Table' (Node 64). To the right, a screenshot of the 'Dialog - 0:65 - SMO (3.7)' configuration window is shown. The 'filterType' dropdown is set to 'No normalization/standardization' and the 'kernel' dropdown is set to 'PolyKernel -C 250007 -E 1.0'. Blue arrows point to these two settings.

Here are the first coefficients of the classification function.

The screenshot shows the 'Weka Node View - 0:65 - SMO (3.7)' window. The 'Weka Output' pane contains the following text:

```
SMO

Kernel used:
  Linear Kernel: K(x,y) = <x,y>

Classifier for classes: acq, crude

BinarySMO

Machine linear: showing attribute weights, not support vectors.

-0.0131 * abm
+ -0.041 * gold
+ 0.0299 * corp
+ -0.029 * proceed
+ 0.0051 * initi
+ -0.0202 * public
+ -0.0543 * offer
+ 0.0072 * seven
+ 0.208 * mln
+ -0.0728 * share
```

Here is the whole workflow under Knime. It is impressive!

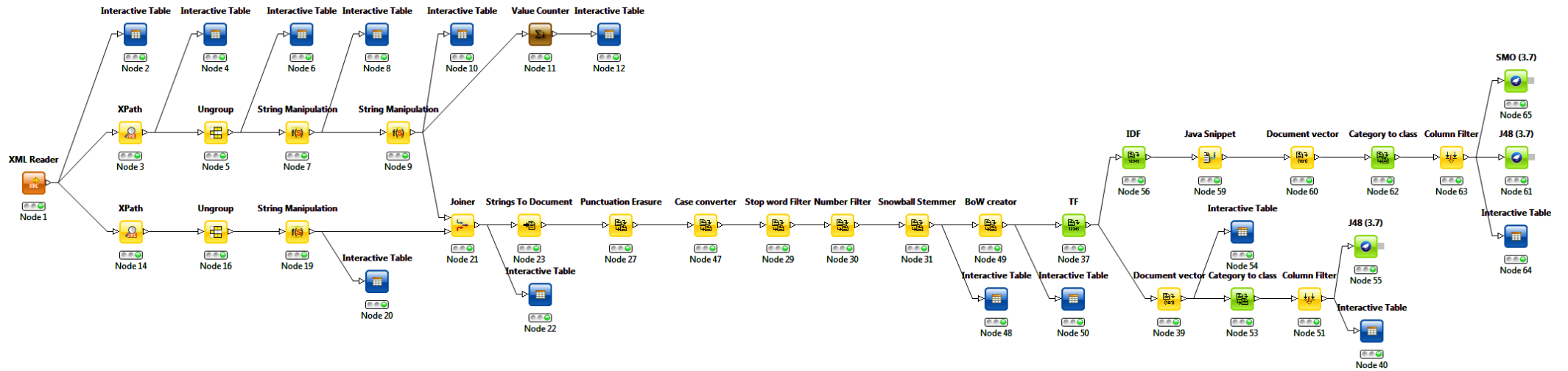


Figure 1 – The whole workflow under KNIME



3.11 Conclusion

I made simple in this tutorial. The classifier is developed based on all the available documents. If we want to go further and get an honest measure of the classifier performance, we must use an evaluation scheme such as cross-validation. Knime can do that easily⁶. In this case, it is important that the generation of the document term matrix must be incorporated into the resampling loop.

4 Document classification using RapidMiner

[RapidMiner Studio](#) is a well-known data mining platform. We use the STARTER version in this tutorial (*available at April 2014, when I wrote the French version of this tutorial*). The performing of the analysis is very easy... when we understand the underlying principle of the organization of the process under RapidMiner. It took me a bit of research on the web before to understand the crucial role of the PROCESS DOCUMENTS FROM DATA component in the process.

4.1 Importation of the documents

RapidMiner can read XML files with the "Read XML" node. I however preferred to use a file in Excel format in this section. The idea is to show that it is finally possible to store textual information in any kind of file, so long as we are able to distinguish the documents, the category (subject) and the plain text (text).

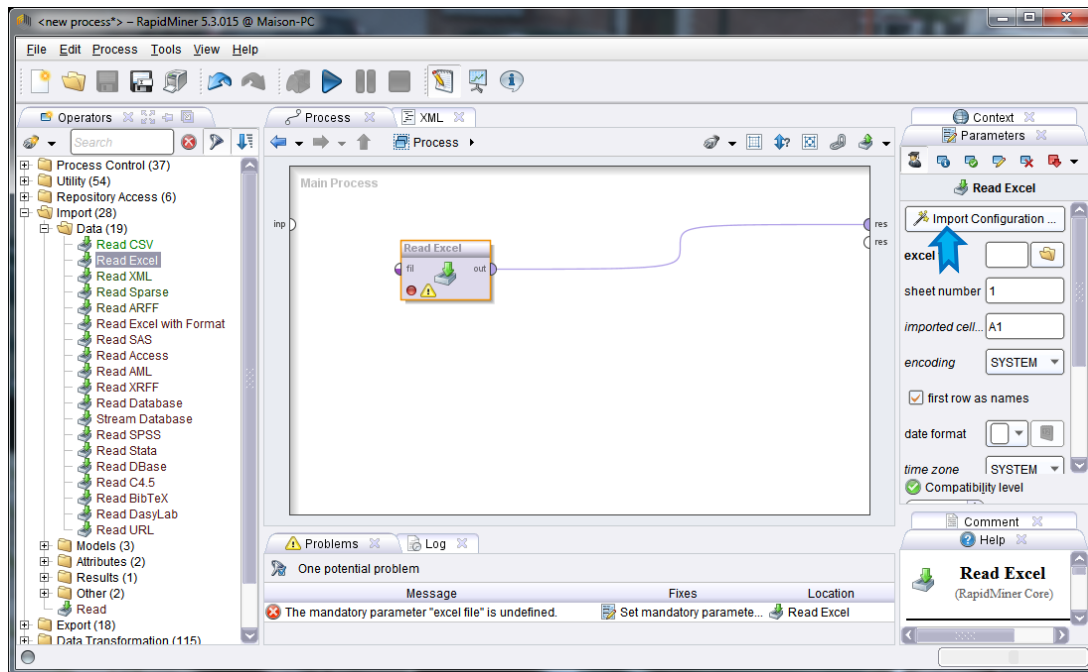


	A	
1	sujet	texte
2	acq	Resdel Industries Inc saidit has agreed to acquire San/Bar Corp in a share-f
3	acq	Warburg, Pincus Capital Co L.P., aninvestment partnership, said it told repre
4	acq	TransCanada PipeLines Ltd deniedreports that it raised its offer for Dome Pe
5	acq	Centel Corp said it completed the sale ofits water properties serving 8,000 cu
6	acq	PBS Building Systems of America Inc,an Anaheim, Calif., company, told the
7	acq	CSR Ltd &CSRA S> intends to proceed withits planned bid for building
8	acq	&Virginia Federal Savings and LoanAssociation> said it has signed a de
9	acq	Allied-Signal Inc said itagreed to sell its Amphenol Products unit to a subsidi
10	acq	Lloyds Investment Managers Ltd, aLondon-based investment firm, said it rais
11	acq	Arthur Appleton, a Chicago investor,told the Securities and Exchange Comm
12	acq	Commonwealth Aluminum(Comalco) said it put its Goldendale, Wash., smelt
13	acq	French state-owned aluminium and specialmetals group Pechiney &PUKG

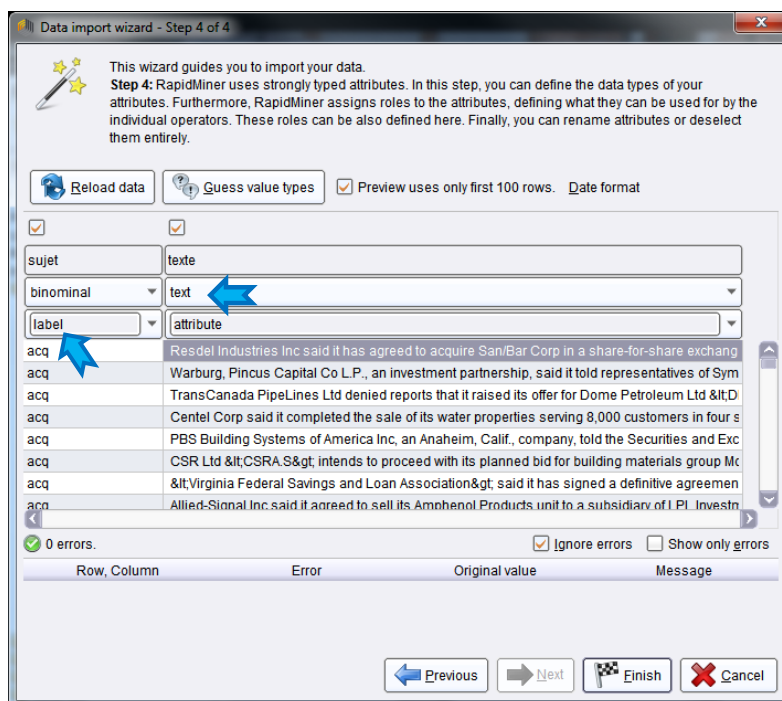
The target variable is the first column of the "reuter.xls" file, the texts are into the second one.

After we launched RAPIDMINER, we create a new "process" (FILE / NEW PROCESS). The GUI and the mode of operation are similar to those of Knime. We use the "Read Excel" node to import the data file. We click on the "Import Configuration Wizard" button.

⁶ <http://data-mining-tutorials.blogspot.fr/2008/11/decision-tree-and-cross-validation.html>

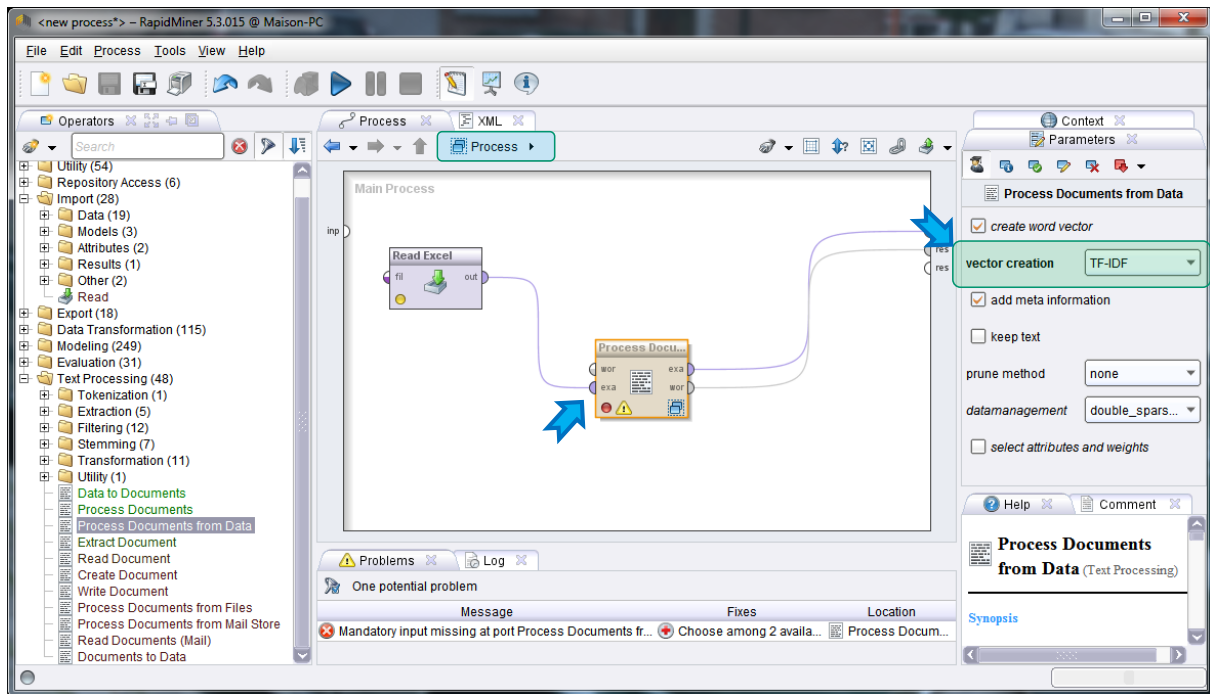


A key step in the import process allows you to specify the role of the columns: “topic” corresponds to the label of the documents; “text” is of type “text”.

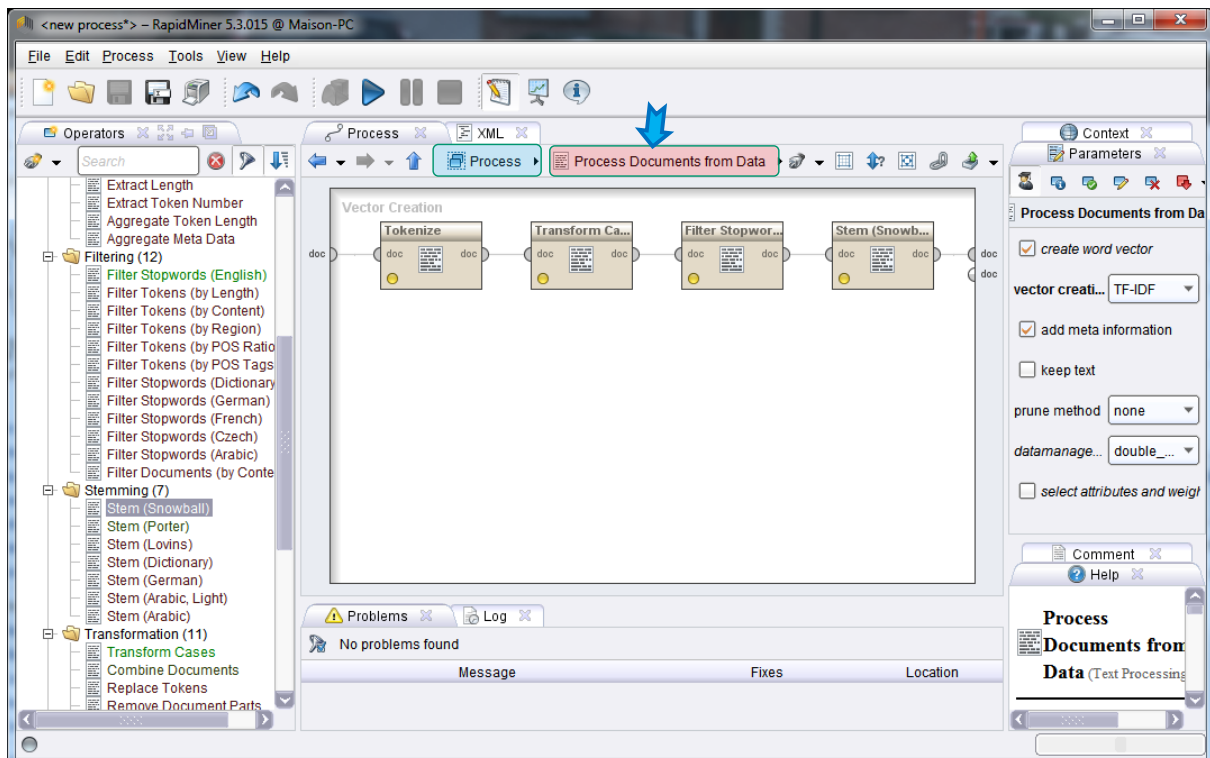


4.2 The « Process Documents from Data » node

We add the tool TEXT PROCESSING / PROCESS DOCUMENTS FROM DATA into the workspace. We connect the “Read Excel” node to the input connection “example set”. The TF-IDF weighting is the default setting.



PROCESS DOCUMENTS FROM DATA is actually a composite tool that does directly generate the matrix documents-terms, by adding the "subject" column since we had taken care to type it as "label" during the importation process above.



We access to the internal structure by double-clicking on the node. We can specify the sequence of treatments for the generation of the document term matrix. We use: TEXT PROCESSING /

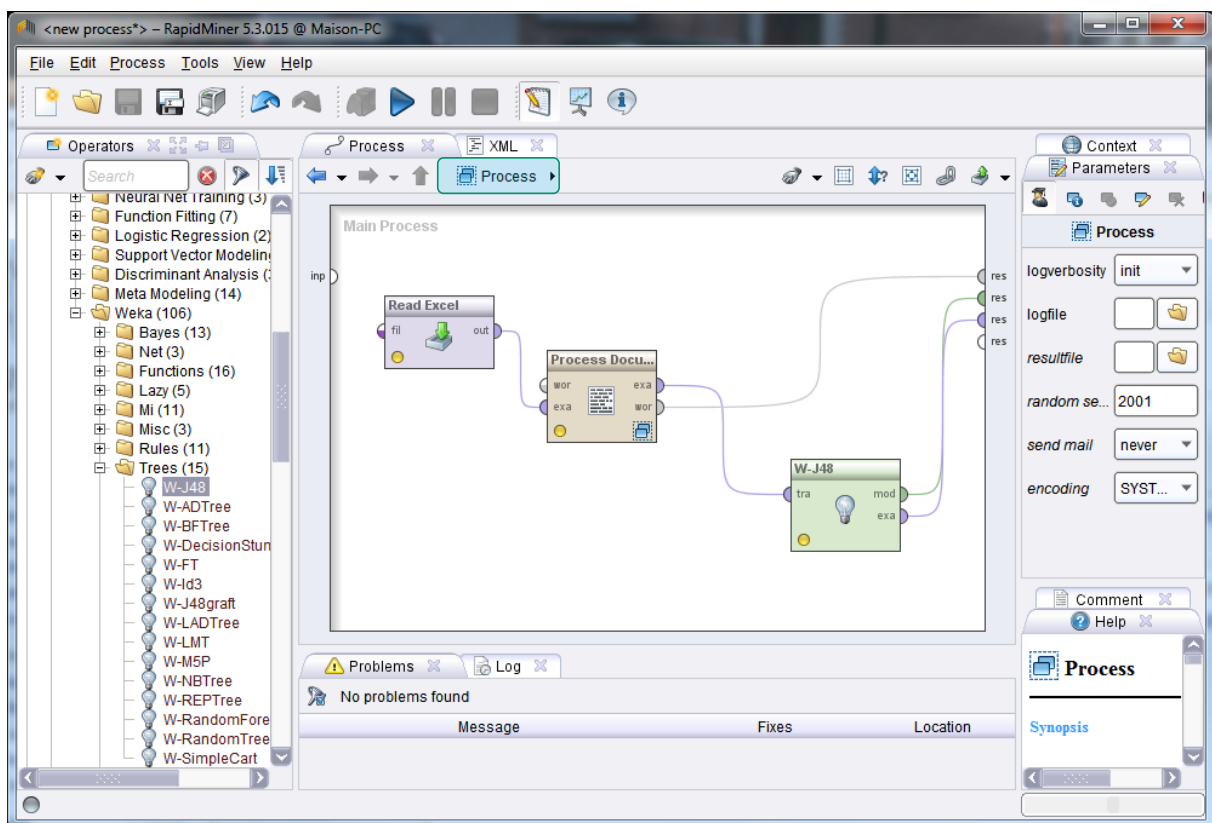


TOKENIZATION / TOKENIZE to identify the words into the text⁷; TEXT PROCESSING TRANSFORMATION / TRANSFORM CASES to change the characters in lower case; TEXT PROCESSING / FILTERING / FILTER STOPWORDS (ENGLISH) to remove the stop words; TEXT PROCESSING / STEMMING / STEM (SNOWBALL) for the stemming operation.

We note that we visualize inside the node PROCESS DOCUMENTS FROM DATA in the screenshot above.

4.3 Machine learning algorithm – Weka J48

We click on the arrow « ↑ » to get back on the main level of the diagram. We insert the node W-J48 imported from the WEKA extension.



We note the different connections, including those that are connected to the output of the diagram. They define the available results at the end of the calculations.

4.4 Results

Now, we can launch the process. A dialog box allows to set the name of the project. We set "Text mining tutorial". Several new tabs containing the results are created.

⁷ <http://en.wikipedia.org/wiki/Tokenization>



Wordlist. The terms are enumerated in this tab. For instance, “accord” is present in 11 documents, it appears 15 times in all, that means it can appear several times in one document. 10 of them are related to the topic “acq”, 5 to “crude”. This information is not trivial. It gives us indications about the relevance of the terms for the prediction of the classes.

The screenshot shows the WordList view in RapidMiner. A blue arrow points to the 'WordList (Process Documents from Data)' tab. The table below shows the data:

Word	Attribute Name	Total Occurrences	Document Occurrences	acq	crude
abegglen	abegglen	2	1	2	0
abil	abil	4	4	3	1
abl	abl	5	5	3	2
abm	abm	3	1	3	0
absolut	absolut	1	1	0	1
ac	ac	1	1	1	0
acceler	acceler	2	2	1	1
accept	accept	4	3	4	0
access	access	2	2	2	0
accord	accord	15	11	10	5
account	account	8	5	3	5
accumul	accumul	3	3	2	1
accur	accur	1	1	1	0

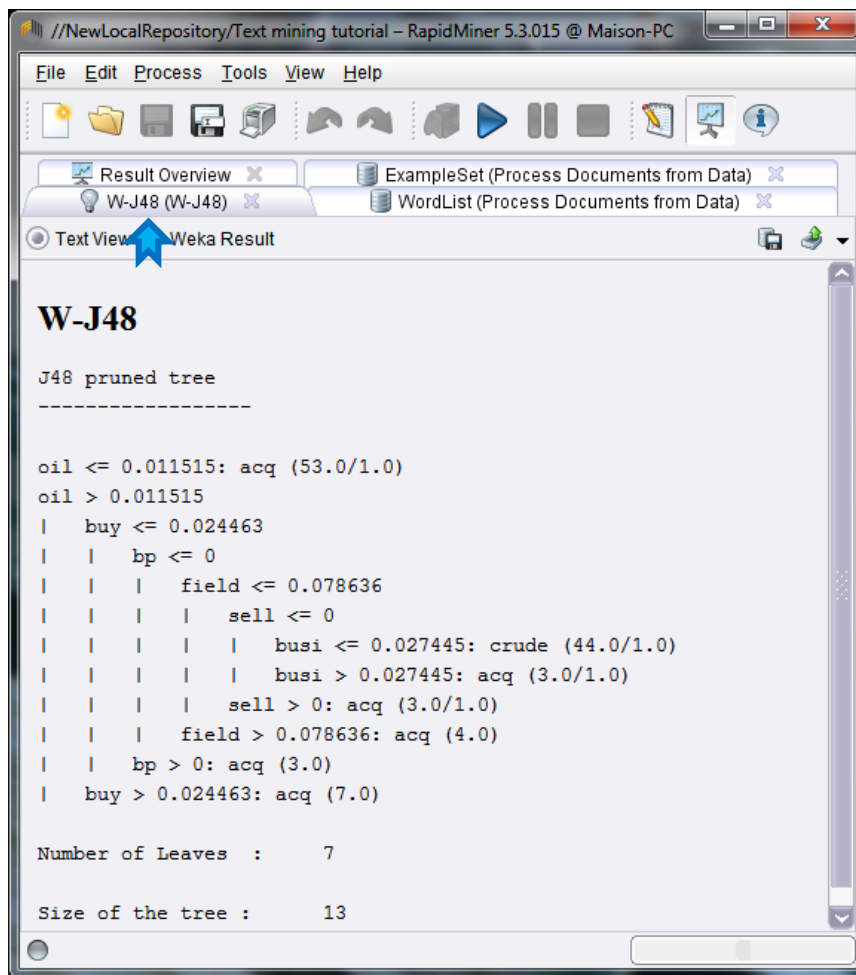
ExampleSet. We visualize the documents-terms matrix. 2329 terms have been generated. The results are slightly different from those of Knime because we have not introduced the same operators of cleanup (e.g. “remove punctuations” under Knime, etc.); because some cleaning algorithms are likely not implemented exactly in the same way (e.g. stemming).

The screenshot shows the ExampleSet view in RapidMiner. A blue arrow points to the 'Data View' tab. The table below shows the data:

Row No.	sujet	abandon	abegglen	abil	abl	abm	absolut	ac	acceler	accept
1	acq	0	0	0	0	0	0	0	0	0
2	acq	0	0	0	0	0	0	0	0	0
3	acq	0	0	0	0	0	0	0	0	0.067
4	acq	0	0	0	0	0	0	0	0	0
5	acq	0	0	0	0	0	0	0	0	0
6	acq	0	0	0	0	0	0	0	0	0.054
7	acq	0	0	0	0	0	0	0	0	0
8	acq	0	0	0	0	0	0	0	0	0
9	acq	0	0	0	0	0	0	0	0	0
10	acq	0	0	0	0	0	0	0	0	0
11	acq	0	0	0	0	0	0	0	0	0



Decision tree. Last, we get the decision tree provided by the J48 algorithm.



The screenshot shows the RapidMiner interface with a 'Weka Result' window. The window title is 'W-J48' and it displays the following text:

```
W-J48

J48 pruned tree
-----

oil <= 0.011515: acq (53.0/1.0)
oil > 0.011515
| buy <= 0.024463
| | bp <= 0
| | | field <= 0.078636
| | | | sell <= 0
| | | | | busi <= 0.027445: crude (44.0/1.0)
| | | | | busi > 0.027445: acq (3.0/1.0)
| | | | | sell > 0: acq (3.0/1.0)
| | | | | field > 0.078636: acq (4.0)
| | | | | bp > 0: acq (3.0)
| | | | | buy > 0.024463: acq (7.0)

Number of Leaves :    7

Size of the tree :    13
```

Despite the fact that we use the same learning algorithm, the tree is different from the one of Knime because the tools have not provided the same document-term matrix. An interesting task would be to compare the list of terms generated by Knime and RapidMiner.

5 Conclusion

The main conclusion of this tutorial that we can perform the document classification process with Knime and RapidMiner. We can perform the same kind of analysis also with my students when we use R and the specialized packages ([tm](#), etc.) during our tutorial classes. The differences lie on the underlying algorithm of the tools used at each step (e.g. when the stemming algorithm is not implemented identically, the lists of stop words are not the same, ...) or on the available parameters that we can set.