Data Mining Methods Week 11 RapidMiner Lab - Text Mining

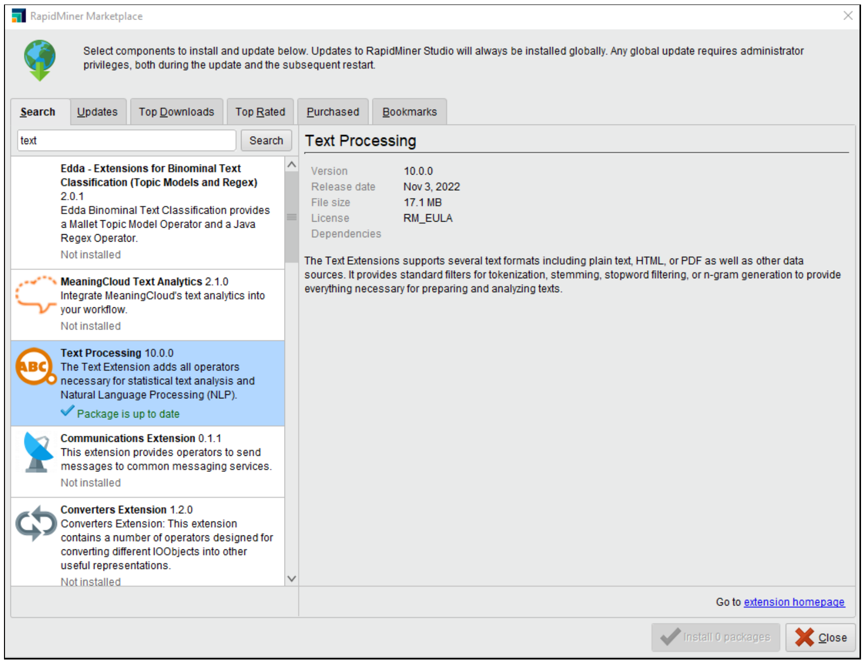
Learning Objectives:

- Understand what text mining is, how it is used, and the benefits of using it;
- Recognize the various formats that text can be in for text mining;
- Develop text mining models in RapidMiner using common text-parsing operators such as tokenization, stop word filtering, n-gram construction, stemming, etc.

Dataset: Please download the dataset titled "SMSSpamCollection.csv" from WT Class to use it with this lab session.

Requirements: After completing this lab, please answer those questions highlighted in yellow and then submit your answers via LA8 Submission.

On the top menu bar, go to Extensions-> Marketplaces (Update and Extensions). In the dialog window, search "text" or go to the "Top Downloads" tap and select the Text Mining Processing to install.



RapidMiner Marketplace

Select components to install and update below. Updates to RapidMiner Studio will always be installed globally. Any global update requires administrator privileges, both during the update and the subsequent restart.

Search | Updates | Top Downloads | Top Rated | Purchased | Bookmarks

text — Search

Edda - Extensions for Binominal Text Classification (Topic Models and Regex) 2.0.1
Edda Binominal Text Classification provides a Mallet Topic Model Operator and a Java Regex Operator.
Not installed

MeaningCloud Text Analytics 2.1.0
Integrate MeaningCloud's text analytics into your workflow.
Not installed

Text Processing 10.0.0
The Text Extension adds all operators necessary for statistical text analysis and Natural Language Processing (NLP).
✓ Package is up to date

Communications Extension 0.1.1
This extension provides operators to send messages to common messaging services.
Not installed

Converters Extension 1.2.0
Converters Extension: This extension contains a number of operators designed for converting different IOObjects into other useful representations.
Not installed

Text Processing

Version	10.0.0
Release date	Nov 3, 2022
File size	17.1 MB
License	RM_EULA
Dependencies	

The Text Extensions supports several text formats including plain text, HTML, or PDF as well as other data sources. It provides standard filters for tokenization, stemming, stopword filtering, or n-gram generation to provide everything necessary for preparing and analyzing texts.

Go to extension homepage

Install 0 packages | Close

Objectives (Business Understanding)

Text message spam is a triple threat: It often uses the promise of free gifts or product offers to get you to reveal personal information; it can lead to unwanted charges on your cell phone bill; and it can slow cell phone performance (FEDERAL TRADE COMMISSION-Consumer Information). Therefore, it is important to predict whether an instant message is spam or not and then to prevent it from sending or spreading to other people. However, text messages are unstructured and it is also a challenge to read each message manually. Text mining provides a good method to analyze the unstructured text in a structured way.

Data Understanding

Different from previous labs, this lab includes the unstructured data. The dataset ''SMSSpamCollection.csv" contains

5,574 SMS (Mobile phone message), among which 747 messages are marked as "spam" and the remainder are non-spam messages, marked as "ham". It is tab-separated text file with one message per line.

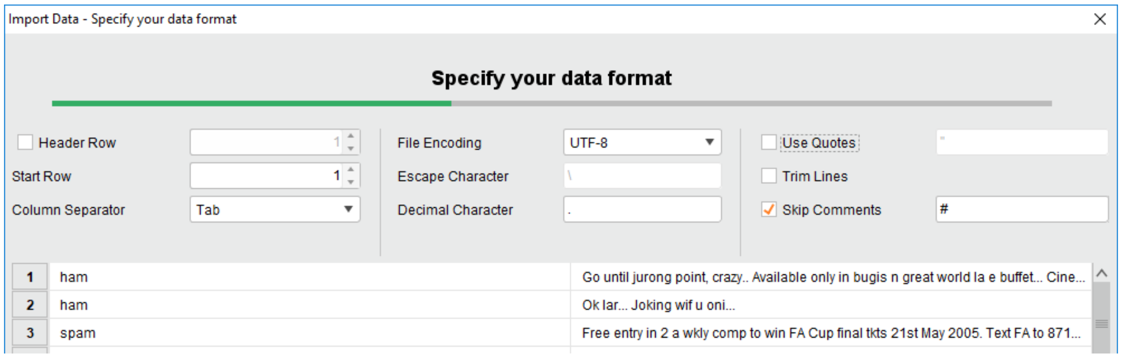
Data Preparation

Texts could be saved in many different formats, so it is a little bit challenging to import them to RM (RM provides many options so that different types of texts could be fed.

1.**Import** the "SMSSpamCollection.csv" dataset to a new blank RapidMiner process.

1.1 In Step 2 of the data import wizard:

- Column separation using "tab" (RM may automatically identify this)
- Uncheck "Use Quotes" (otherwise, it shows you four errors because of quotes in the messages)
- Uncheck "Header Row" because the first row is not attribute name



Import Data - Specify your data format

Specify your data format

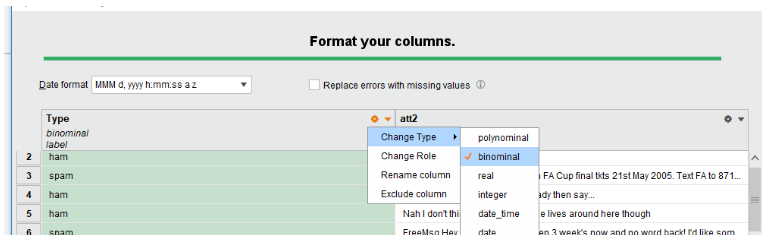
☐ Header Row	1	File Encoding	UTF-8	☐ Use Quotes	"
Start Row	1	Escape Character	\	☐ Trim Lines	
Column Separator	Tab	Decimal Character	.	✓ Skip Comments	#

1	ham	Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine...
2	ham	Ok lar... Joking wif u oni...
3	spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 871...

- Change "File Encoding" to UTF-8. Note: UTF-8 is a compromise character encoding that can be as compact as ASCII (if the file is just plain English text) but can also contain any unicode characters (with some increase in file size). UTF stands for Unicode Transformation Format. The '8' means it uses 8-bit blocks to represent a character.

1.2 In Step 3 of the data import wizard:

- Double click the first column and change the column name from "att1" to "Type"
- Then, change its type to binominal (see the screenshot as below)



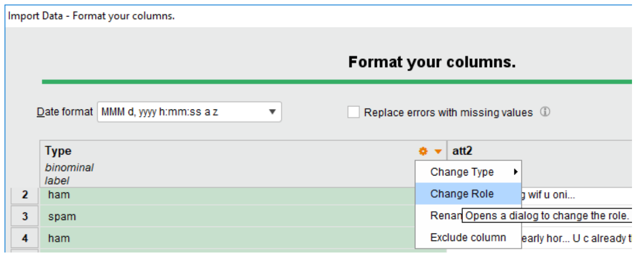
Format your columns.

Date format: MMM d, yyyy h:mm:ss a z ☐ Replace errors with missing values

Type (binominal, label) | att2

2	ham	
3	spam	FA Cup final tkts 21st May 2005. Text FA to 871...
4	ham	dy then say...
5	ham	Nah I don't thi... e lives around here though
6	spam	FreeMsg Hey... en 3 week's now and no word back! I'd like som

- Then, change its role to "label"



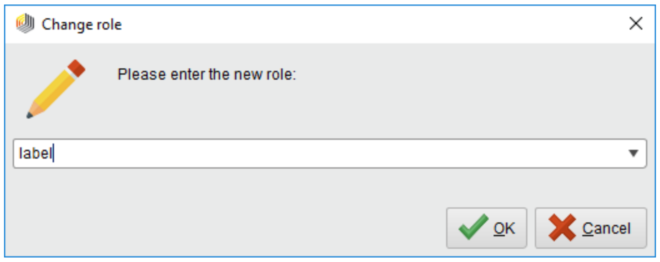
Import Data - Format your columns.

Format your columns.

Date format: MMM d, yyyy h:mm:ss a z ☐ Replace errors with missing values

Type (binominal, label) | att2

2	ham	wif u oni...
3	spam	
4	ham	early hor... U c already t



Change role

Please enter the new role:

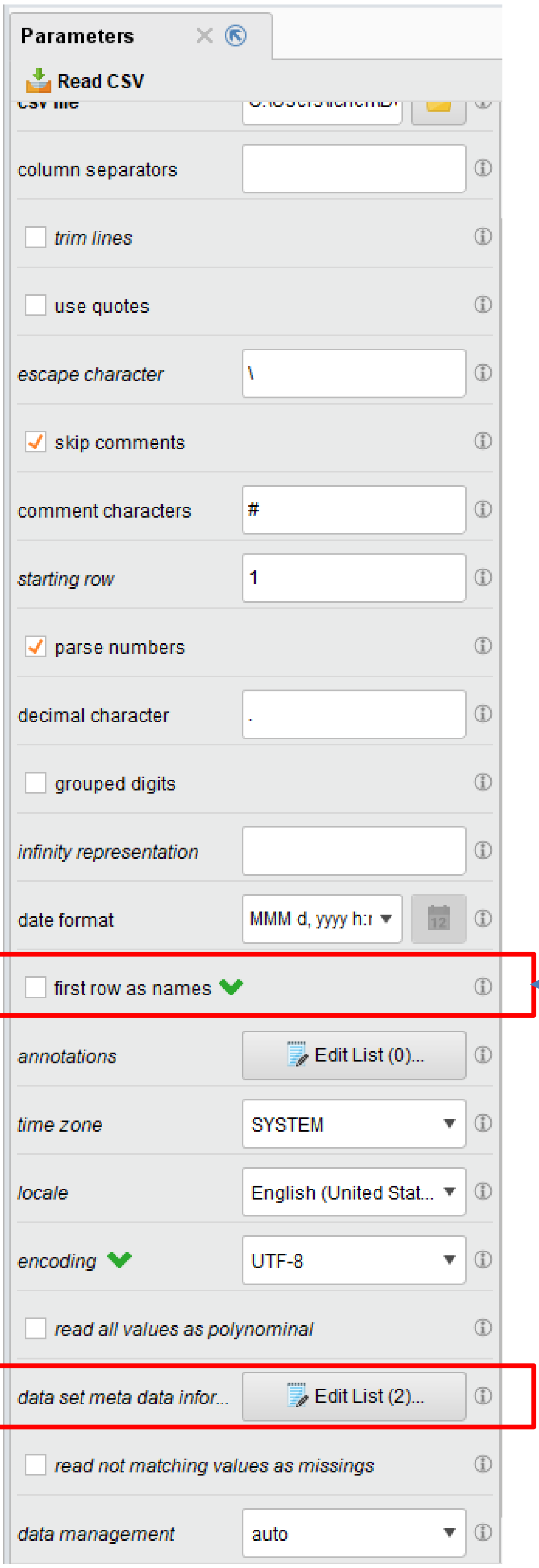
label

- Double click the second column and change the column name from "att2" to "Message"

Note: in the new version of RM, it does not offer a choice to change the data type of the second attribute from polynominal to text (this function is available in the earlier version of RM), so we will do this later.

1.3 Click "Finish"

1.4 Please check the parameters of the Read CSV operator and then change the data type of Message to Text by following the procedure in the next page.



Parameters

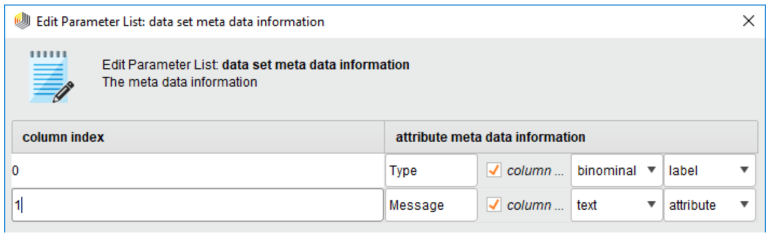
Read CSV

column separators	
☐ trim lines	
☐ use quotes	
escape character	\
✓ skip comments	
comment characters	#
starting row	1
✓ parse numbers	
decimal character	.
☐ grouped digits	
infinity representation	
date format	MMM d, yyyy h:r
☐ first row as names	
annotations	Edit List (0)...
time zone	SYSTEM
locale	English (United Stat...
encoding	UTF-8
☐ read all values as polynominal	
data set meta data infor...	Edit List (2)...
☐ read not matching values as missings	
data management	auto

Make sure this is unchecked; otherwise, you will only get 5573, rather than 5574 records (the similar issue occurred in HW1)

Change the data type of the second attribute to Text

1. Click Edit List (2) for the data set meta data (recall this concept in Week 2)
2. Change their meta data as below and then click Apply

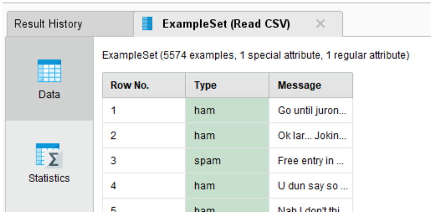


Edit Parameter List: data set meta data information

Edit Parameter List: **data set meta data information**
The meta data information

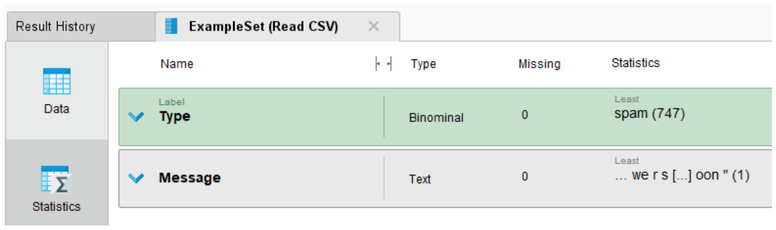
column index	attribute meta data information			
0	Type	✓ column ...	binominal	label
1	Message	✓ column ...	text	attribute

1.5 Save this process as Lab11_Step 1 and then run it. You are supposed to get the following results in your data view and Statistics view. If not, please go back to check if you follow all the instructions in Steps 1.1-1.4.


Result History | ExampleSet (Read CSV)

ExampleSet (5574 examples, 1 special attribute, 1 regular attribute)

Row No.	Type	Message
1	ham	Go until juron...
2	ham	Ok lar... Jokin...
3	spam	Free entry in ...
4	ham	U dun say so ...
5	ham	Nah I don't thi


Result History | ExampleSet (Read CSV)

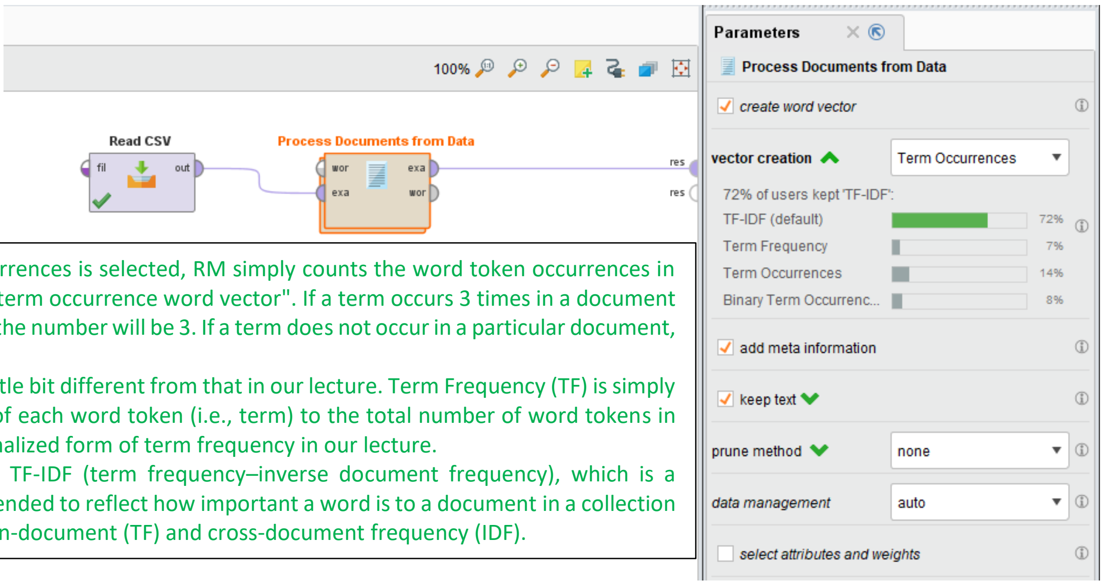
Name	Type	Missing	Statistics
Label Type	Binominal	0	Least spam (747)
Message	Text	0	Least ... we r s [...] oon " (1)

1.6 Take a look at Data view with date and time and then manually check a few spam messages. Please identify at least two characteristics of spam messages [No question to answer].

1.7 Take a look at your Statistics view with date and time. What is the percentage of spam messages out of all the messages? What are the top three most frequent messages? [Note: Here, RM has not combine all the similar messages].

2. Parse the messages into words and build the term document matrix.

2.1 We use the Process Document from Data Operator, which takes a text file as input and create a term document matrix as output. We use "Term Occurrences" as the metrics for terms. Check "create word vector", "add meta information", and "keep text".



Parameters

Process Documents from Data

✓ create word vector

vector creation: Term Occurrences

72% of users kept 'TF-IDF':

TF-IDF (default)	72%
Term Frequency	7%
Term Occurrences	14%
Binary Term Occurrenc...	8%

✓ add meta information

✓ keep text

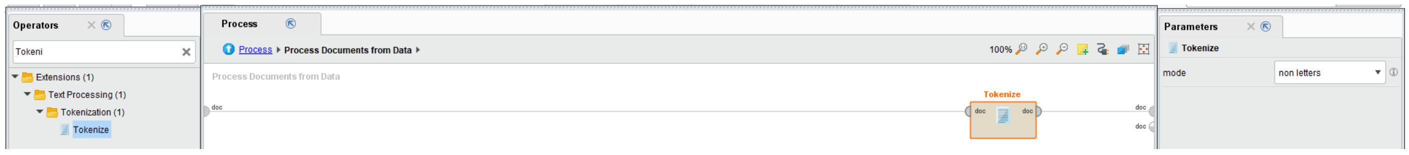
prune method: none

data management: auto

select attributes and weights

Side Note: When Term Occurrences is selected, RM simply counts the word token occurrences in each document to create a "term occurrence word vector". If a term occurs 3 times in a document (i.e., a message in this case), the number will be 3. If a term does not occur in a particular document, its value will be 0.
Term Frequency in RM is a little bit different from that in our lecture. Term Frequency (TF) is simply the ratio of the occurrence of each word token (i.e., term) to the total number of word tokens in the document. This is a normalized form of term frequency in our lecture.
The most popular choice is TF-IDF (term frequency–inverse document frequency), which is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It considers both in-document (TF) and cross-document frequency (IDF).

2.2 Inside the Process Document from Data Operator, we use the Tokenize operator, which extracts all individual words from the text file. Use non letters as the mode (This selects the tokenization mode. Depending on the mode, split points are chosen differently). Save your process as Lab11_Step2.2 and then run it.



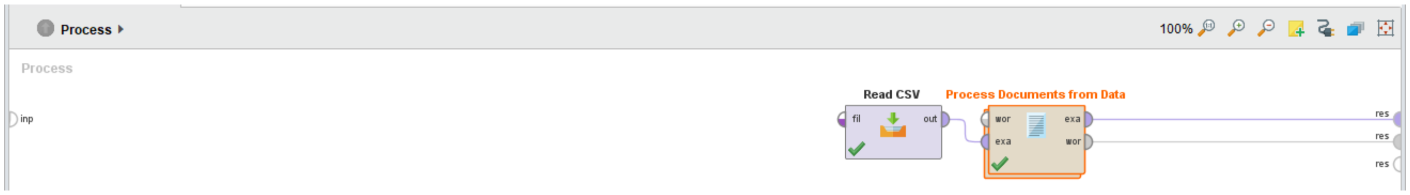
Parameters

Tokenize

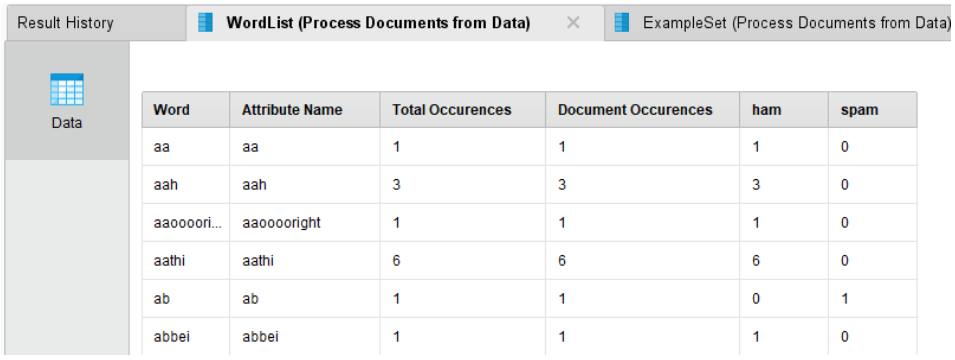
mode: non letters

2.3 Take a look at the Data view of your result with date and time. Check the data view and answer the following questions: How many rows and columns do you have? What does a row in the result mean? What does a column in the result mean? (Hint: in Step 2.1, we use Term Occurrences to create the vector).

2.4 Go back to the design mode and add a connection between the wor port and the res port (see the screenshot below) to generate a word list.



2.5 Run this process and take a look at the word list as below.


Result History
WordList (Process Documents from Data)
ExampleSet (Process Documents from Data)
Data

Word	Attribute Name	Total Occurences	Document Occurences	ham	spam
aa	aa	1	1	1	0
aah	aah	3	3	3	0
aaoooori...	aaooooright	1	1	1	0
aathi	aathi	6	6	6	0
ab	ab	1	1	0	1
abbei	abbei	1	1	1	0

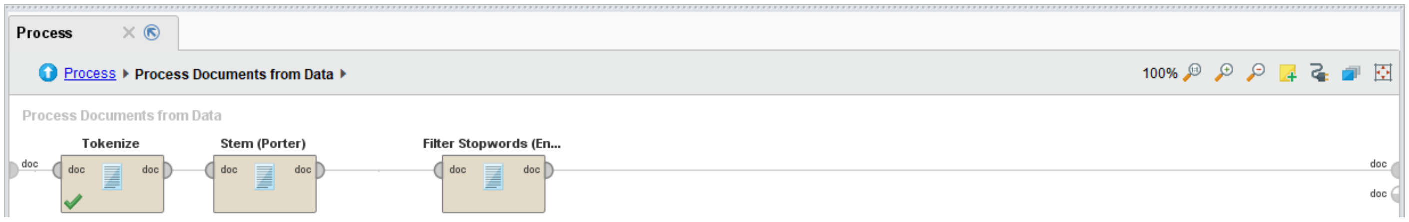
2.6 Please sort by Total Occurrences at a descending order and then take a look at the sorted word list. What are the top three words in all the messages? Then, compare the top three words' occurrences in ham and spam (the final two columns) and indicate which word(s) occur(s) more frequently in one type than the other. Hint: Please also consider the number of ham and spam messages in our dataset. Next, sort the wordlist by ham and spam respectively, and find the top three words in each list (ham and spam messages).

3. Stem, Filter stop words, and add N-gram

3.1 Add the operator Stem (Porter) into the Process Document from Data operator. This operator stems English words using the Porter stemming algorithm applying an iterative, rule-based replacement of word suffixes intending to reduce the length of the words until a minimum length is reached.

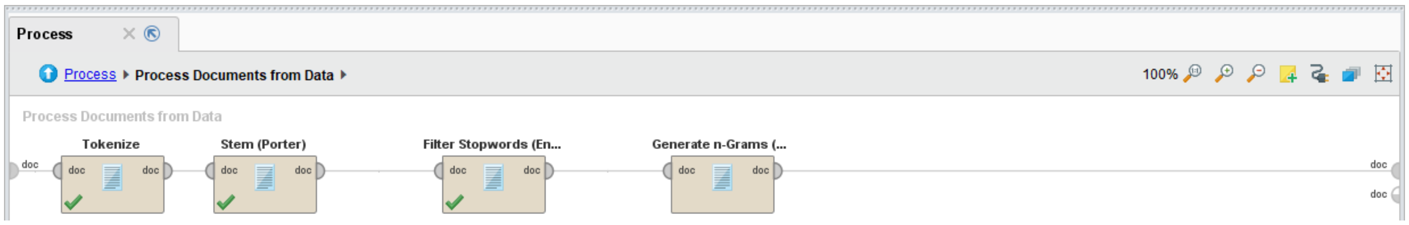
3.2 Add the Filter Stopwords (English) operator, which filters English stopwords from a document by removing every token which equals a stopword from the built-in stopword list. Please note that, for this operator to work properly, every token should represent a single English word only. To obtain a document with each token representing a single word, you may tokenize a document by applying the Tokenize operator beforehand.

3.3 Save this process as Lab11_Step3.3 and then run it.



3.4 Take a look at your ExampleSet (Process Documents from Data) and check how many columns in your ExampleSet you have now. Please think about why you have fewer columns now.

3.5 Add the Generate N-Gram (Terms) Operator. Set Max Length as 2. Save this process as Lab11_Step 3.4 and then run it (It takes time to run this process).



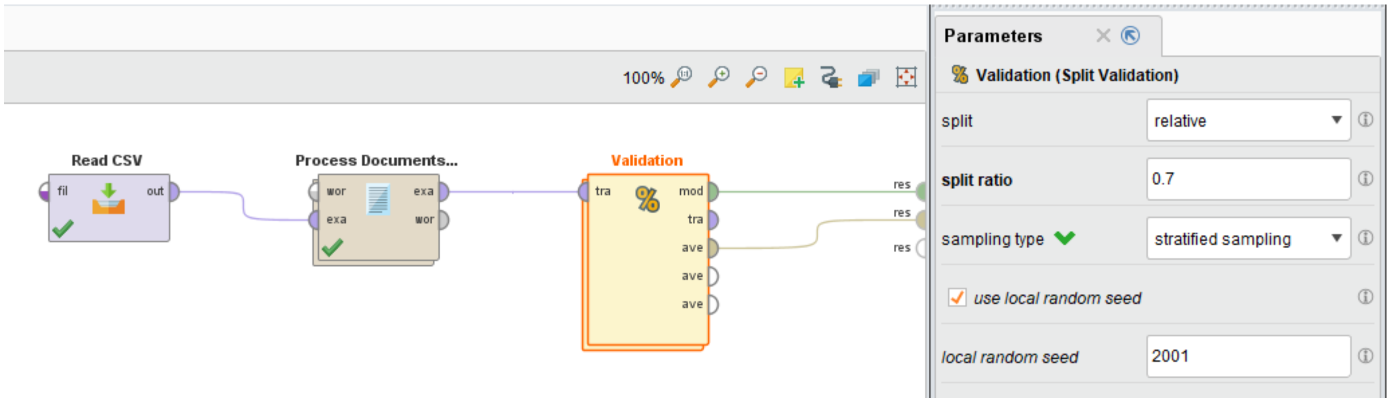
Note: Generate N-Gram (Terms)": This operator creates term n-Grams of tokens in a document. A term n-Gram is defined as a series of consecutive tokens of length n. The term n-Grams generated by this operator consist of all series of consecutive tokens of length n.

3.6 Take a look at the ExampleSet (Process Document from Data) and check how many columns you have now. Please think about why you have more columns now.

Modeling & Evaluation

We will use Naïve Bayes method to classify messages into spam and ham, and then evaluate the model's performance. This is similar to what we did in Week 6 Lab.

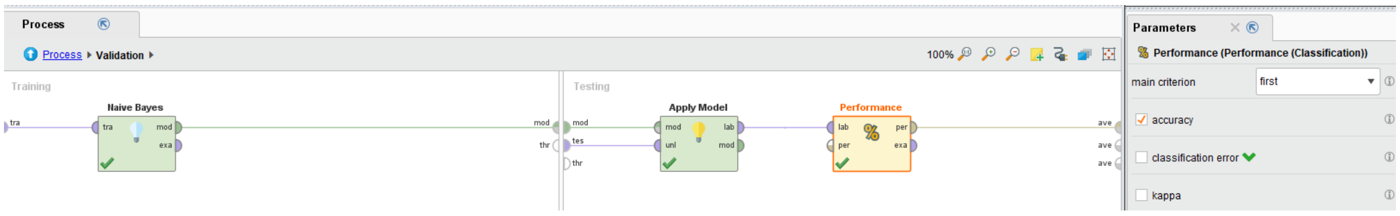
4. We use the operator Split Validation and set sampling type as "stratified sampling". Check "use local random seed and set it as 2001. Note: we did not specify target attribute using Set Role because in Step 1 we set Type as the label, which will be identified as the target attribute.



Validation (Split Validation)

split	relative
split ratio	0.7
sampling type	stratified sampling
✓ use local random seed	
local random seed	2001

5. Inside the operator Split Validation, we use Naïve Bayes, Apply Model and Performance (Classification) operators. Save your process as Lab11_Step 5. Run the process (it takes a while because both text processing and Naïve Bayes are quite computationally expensive).

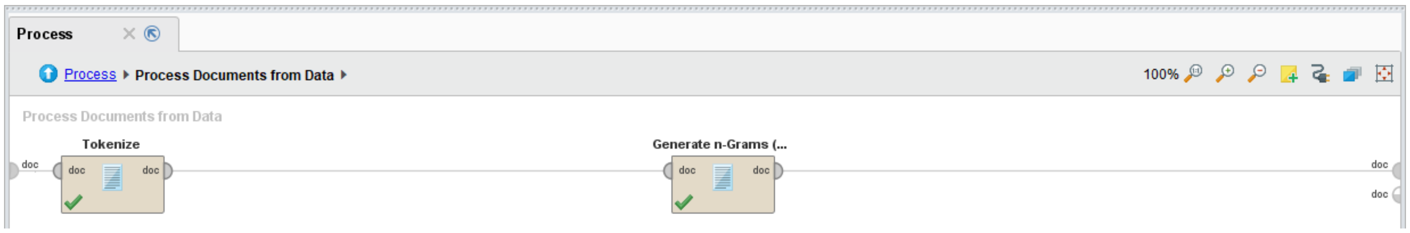


Performance (Performance (Classification))

main criterion	first
✓ accuracy	
classification error	
kappa	

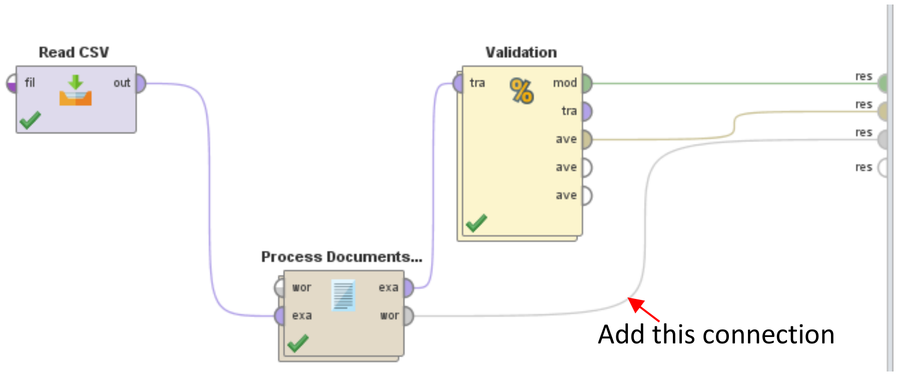
6. Take a look at the confusion matrix. Our interest is identifying spam messages here. Please obtain accuracy, precision, and recall (You may refer back to the Appendix of Week 6 Lab).

7. **Remove** operators you added in Step 3.1 and 3.2. The inside of Process Documents from Data operator should look like the following screenshot. Save this process as Lab11_Step6 and then run it.

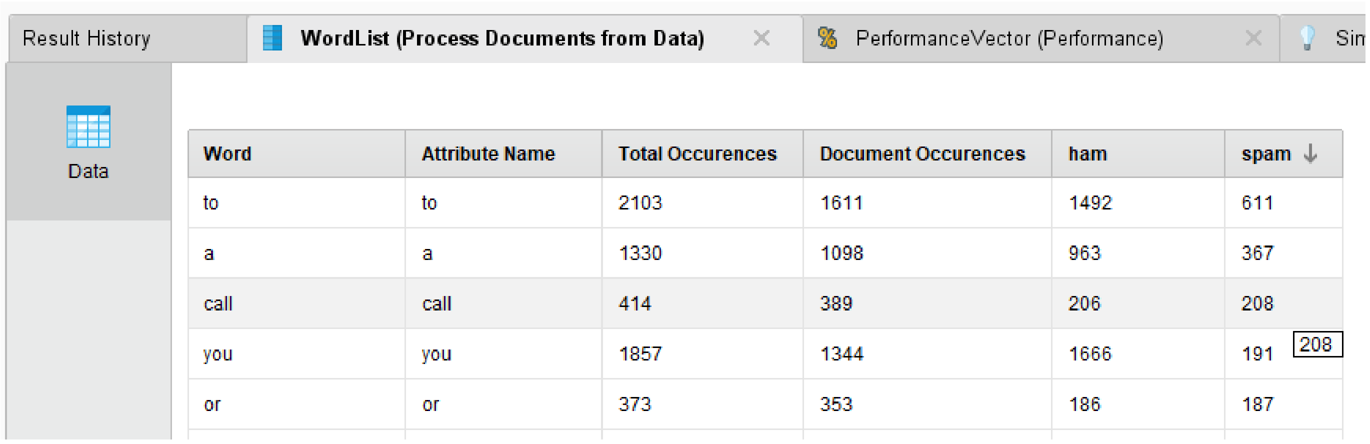

Process
Process ▸ Process Documents from Data ▸
100%
Process Documents from Data
Tokenize
Generate n-Grams (...
doc
doc
doc
doc
doc
doc
doc
doc

8. Take a look at the confusion matrix. Compare the performance with what you got in Step 6. Which one performs better for identifying spam message? Why? (Hint: stop words and variations of same-stem words may mean something here in the spam case).

[Optional] Go back to your process in the design mode, and add a connection between the "wor" port and the result (see below). By doing so, you are going to generate a new word list report.



Please sort the word list by Spam (see the picture as below). You might find that some stop words such as "to" and "a" occur more frequently in spam than in ham. This will further help you answer the question above.


Result History
WordList (Process Documents from Data)
PerformanceVector (Performance)
Sin
Data

Word	Attribute Name	Total Occurences	Document Occurences	ham	spam ↓
to	to	2103	1611	1492	611
a	a	1330	1098	963	367
call	call	414	389	206	208
you	you	1857	1344	1666	191 208
or	or	373	353	186	187