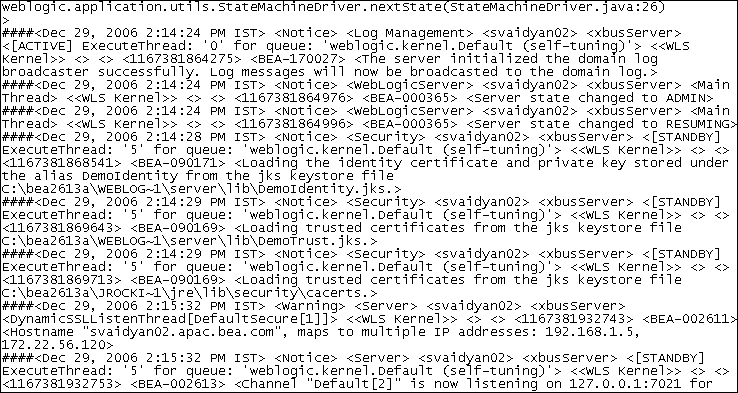
*Start a new Python project folder … do not reuse the previous workspace … As with any keyboard-driven console-like environment, developing muscle -memory for the common commands is also part of the learning curve.*

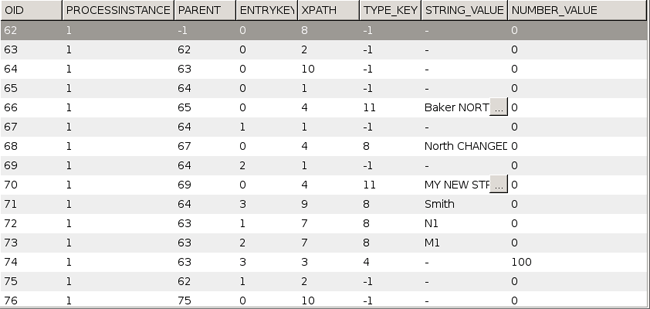
**This Is A Typical Machine Learning Pipeline**

The most likely scenario is that:

***Raw text*** - you will most likely receive a data model that can’t distinguish words, plenty of noises, unstructured, and no consistent delimiters.



***Tokenize*** - parse the model and tell it what to look at to impose some form of a structure. You can do this in several ways including the use of split() function to convert a sentence into a list of words. But some words are more important than other words such as stop words.



***Clean text*** - is the step to filter out (massage, scrub, sanitize, remove stop words/punctuations) stemming, etc. Stemming is part of deep learning to differentiate for example learn, learned, and learning.



***Vectorize*** - is the process of adding some form of intelligence in the data you are interested in to give some meaning to Python. You do this by converting the data into numeric form. In other words, you convert the text into the numeric representation of that text. You are counting the occurences of each word in each text message using a matrix with one row per text message, and one column per word.



***Machine learning algorithm*** - allows you to fit and train your vectorized data to find relationships between the words and the labels in order to make predictions whether an incoming mail is spam or not, for example.



***Spam filter*** - once your model is trained and fit, you can try it with real data before putting your algorithm to production. An example NLP would be a spam filter.

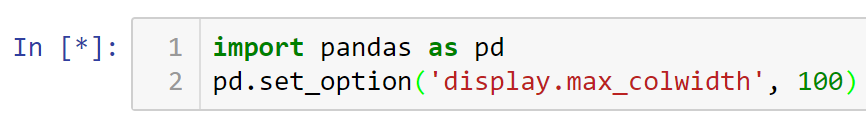


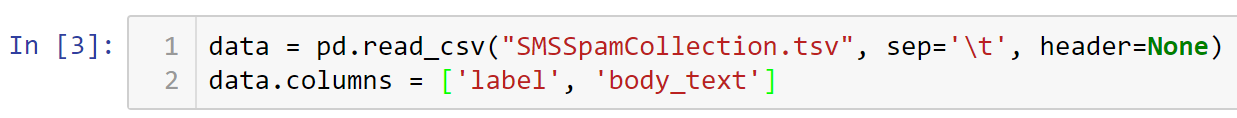
**Pre-processing (cleaning up) text data**

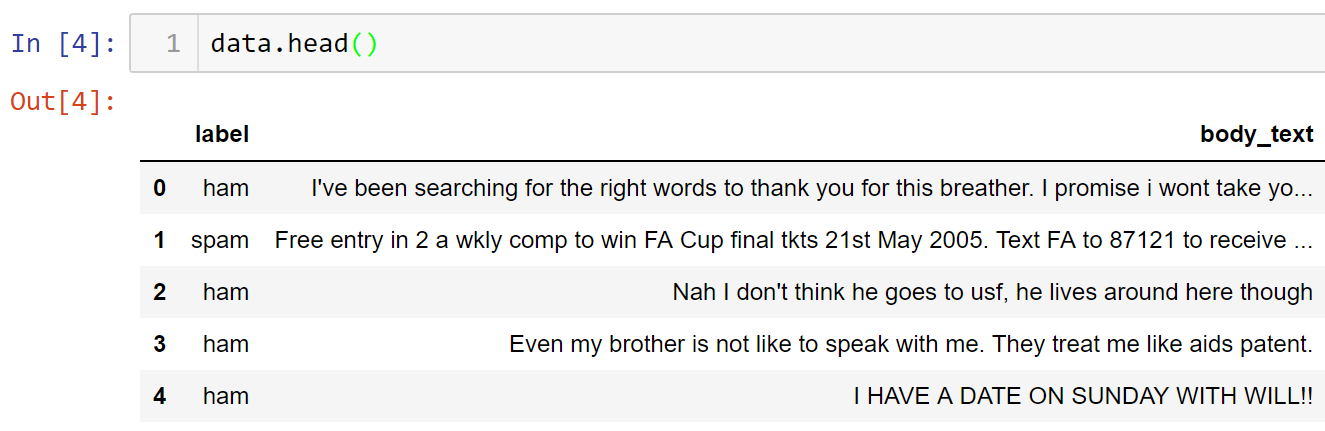
Cleaning up the text data is necessary to highlight attributes that you’re going to want your machine learning system to pick up on. Cleaning (or pre-processing) the data typically consists of a number of steps:

1. Remove punctuation
2. Tokenization
3. Remove stopwords
4. Lemmatize/Stem

The first three steps are extremely important and are covered here and the previous exercises as they are implemented in pretty much any text cleaning pipeline. Lemmatizing and stemming are covered in the next sessions as they are helpful but not critical because they are advanced are are not included in the all machine learning pipeline.

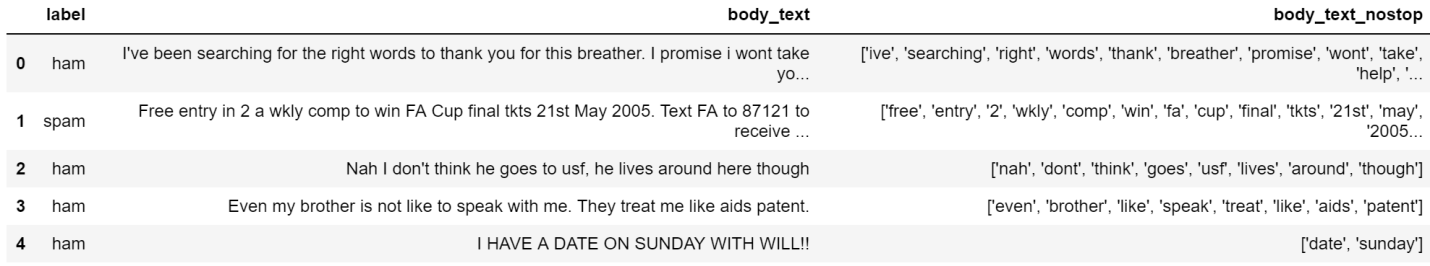






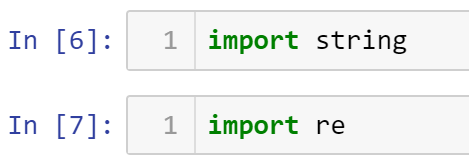
To help you understand where we are going with this, run the code below to see the finish product …



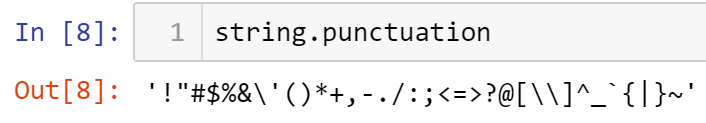


‘body\_text’ column is the raw data that you just did, while ‘body\_text\_nostop’ will be the goal. Note that the cleanup version are tokenized, no punctuations, and stop words removed. Compare the difference for the first row, for example. Now that you have a decent idea of where we are going, let us remove punctuations.

**Removing punctuations**



The string package has a list of punctuations that we can use to tell Python what they look like. ‘re’ package is the regular expression library we used earlier to perform tokenization. To show a list of punctuation ...



For humans, “I like NLP.” is the same as “I like NLP” without the period. But in Python … the two strings are not the same as indicated by a boolean False output below.



So, let’s create a function that will discard punctuations. Building the list comprehension …



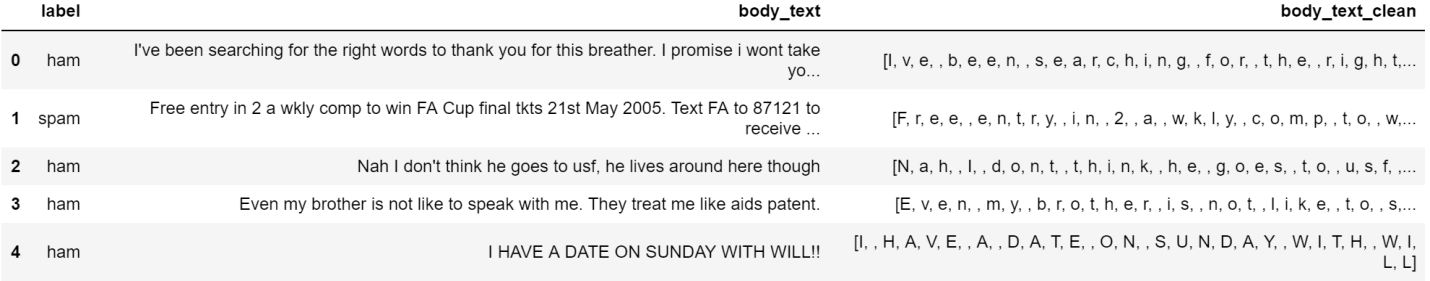
Return the character for each character in the input text for as long as the character is not one of the punctuation.

To use the newly created function, we have to use lambda function to apply it to the body\_text column.

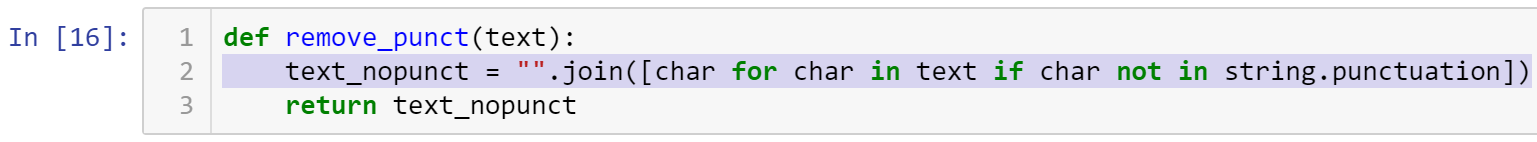


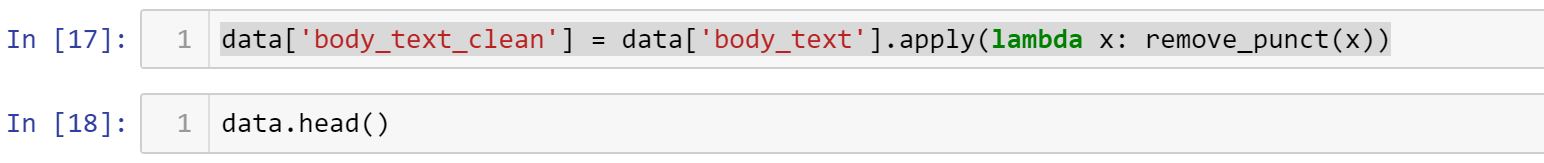
Displaying data …

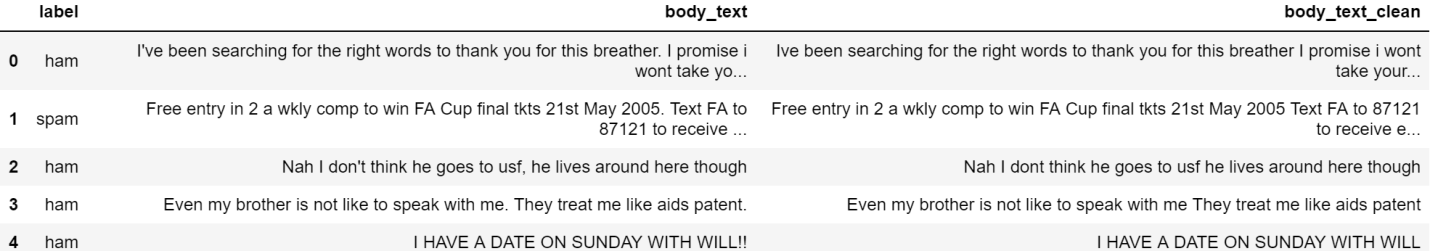




Note in the body\_text\_clean column that the punctuations are eliminated. The result is really not what we wanted because no tokens are formed. A join() function will be use to form the token(or word). Joining the characters will recreate the original sentence without the punctuations. Join these characters with nothing as indicated by “”. Run the tree lines again.





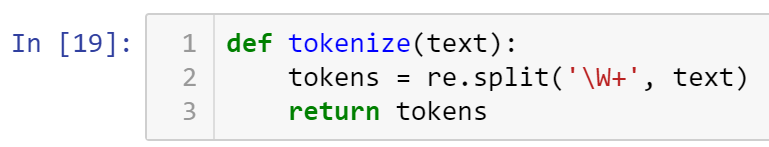


Note that the characters are now joined together as a sentence.

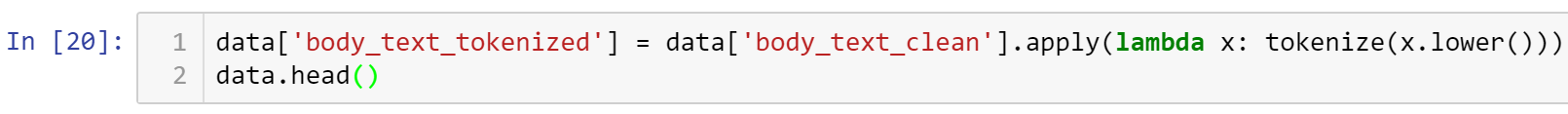
So, that’s how you remove punctuations from raw data.

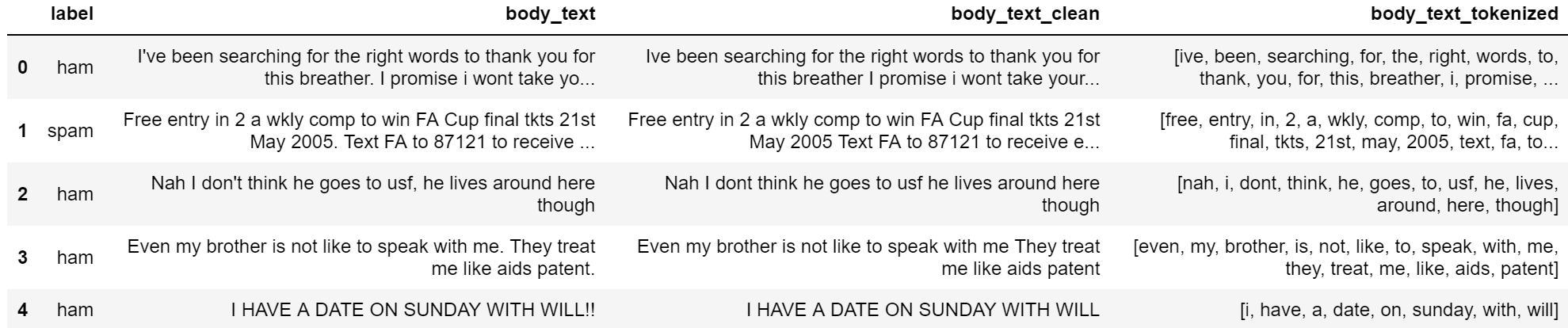
**Tokenization**

Remember that in regex, ‘\W+’ means it will split when it sees one or more non-word character. This include white spaces, special characters, etc.



Applying this function to a column using lambda function … and printing the first few rows ...



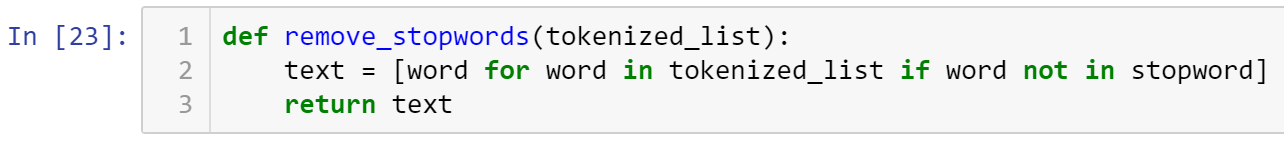


**Removing stopwords**

A list of stop words in English version is available in the nltk library



Create a function that tokenize, and pass the cleaned version of the raw data. Go through all the words, discard stop words, and keep the other words in a list called text.

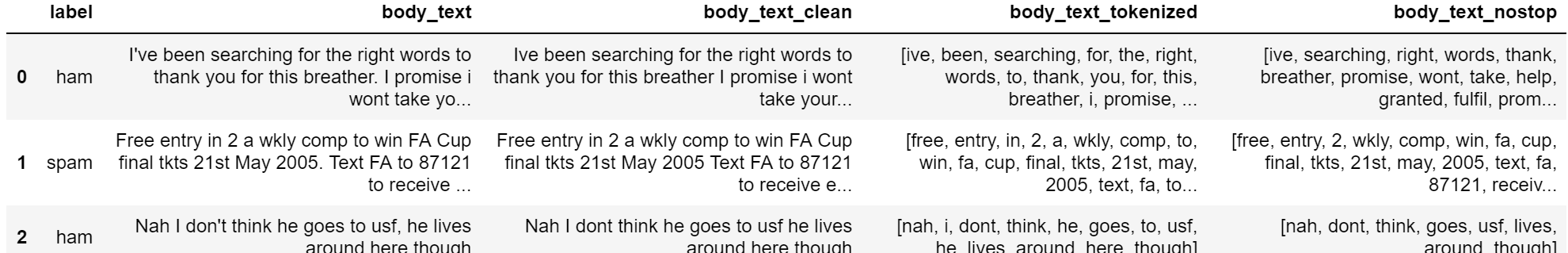


Apply this function to the clean data column using lambda function.



Displaying the few rows …





Note on the first row under body\_text\_tokenized, the words been, for, the, to, you, this, and i are now gone on the new list. Now, we have a clean column that have been tokenized, punctuations are gone, and no stop words.

So, this exercise is an abbreviated look at what a pipeline would look like. You take a raw text and clean it up so deeper learning can be applied to the new model.

* All submissions should be separate from other exercises and quests. Please do not lump all your answers into one document and re-using that same workspace to gain multiple points. Thanks.
* Place your name at the bottom of your code, download your Python program in html format, and submit your work in Canvas.