*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.*

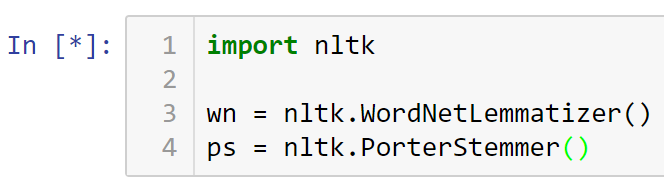
**Lemmatizing** - is the process of grouping together the inflected forms of a word so they can be analyzed as a single term, identified by the word’s lemma (the canonical form of a set of words). For example, type, typed, and typing will all be the same form of the same lemma.

Simply put, lemmatizing is using vocabulary analysis of words aiming to remove inflectional endings to return the dictionary form of a word. So, for example, type, typed, and typing will return type because that is the root of the word.

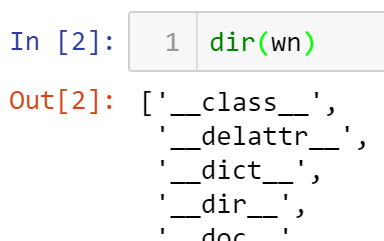
How is lemmatizing different from stemming? They both aim the same process, but slightly different.

* The goal of both is to condense derived words into their base forms
* The difference is that stemming is typically faster as it simply chops off the end of a word using heuristics, without any understanding of the context in which a word is used.
* Whereas lemmatizing is typically more accurate as it uses more informed analysis to create groups of words with similar meaning based on the context around the word.
* Lemmatizing will always return a dictionary word, and, therefore, it is more accurate and required more computationally expensive.
* Stemming use the algorithmic approach, so it only concern with the string it was given by chopping off the suffix. Lemmatizing is a little more complex in that it searches the corpus to find related words. The problem is that if this word is not in the corpus, then it will just return the original word.

The most popular lemmatizer is WordNet. It is a collection of noun, verbs, adjectives, and adverbs that are group together and sets of synonyms, each expressing a distinct concept.

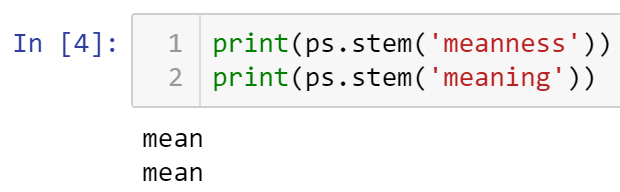


Listing WordNet’s members …

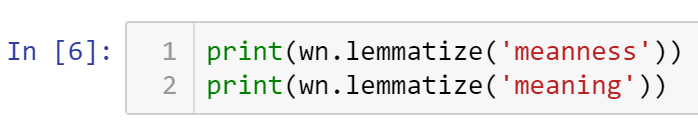


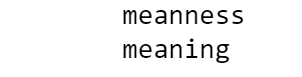
One of its method is lemmatize()

Note that stem is not accurate; for example, the two words below that have different meaning is converted to the same word.



Using lemmatize() …

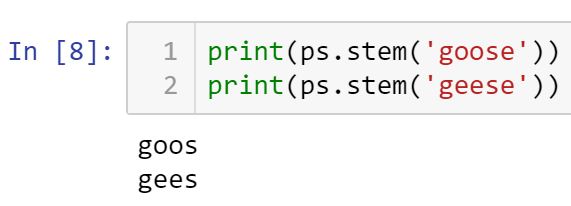




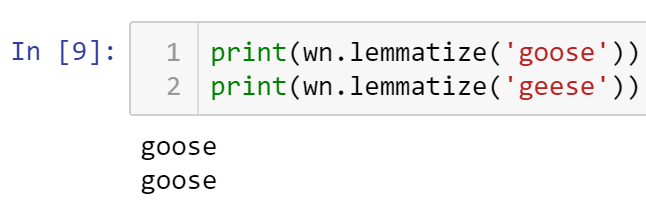
Note the short delay. Also, note that the output are more accurate.

Stemming use the algorithmic approach, so it only concern with the string it was given by chopping off the suffix. Lemmatizing is a little more complex in that it searches the corpus to find related words. The problem is that if this word is not in the corpus, then it will just return the original word. And that is what is happening above. Not condensing it is probably better than translating it with inaccurate meaning.

Let’s look at one more example …



The stemmer does not know what to do here even though one word is just a plural of the other.



The lemmatizer correctly handled the two words above.

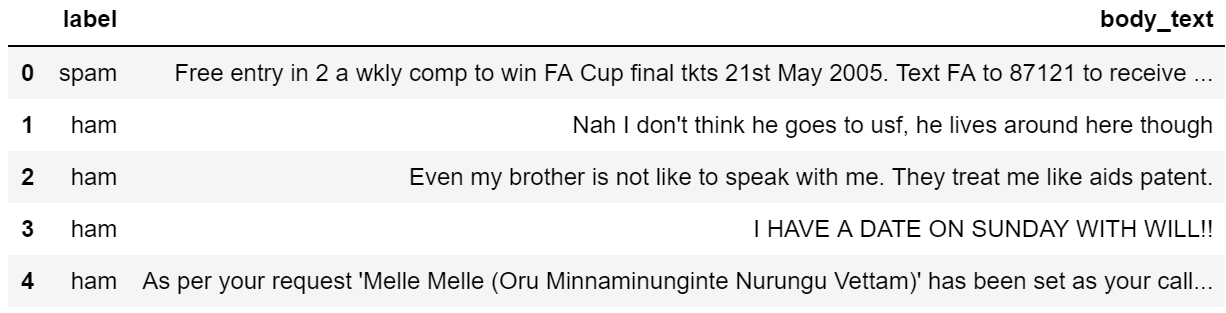
So, lemmatizer is quite powerful in some very complex situations.

Now that we know how lemmatizer works, let’s apply it to our dataset.

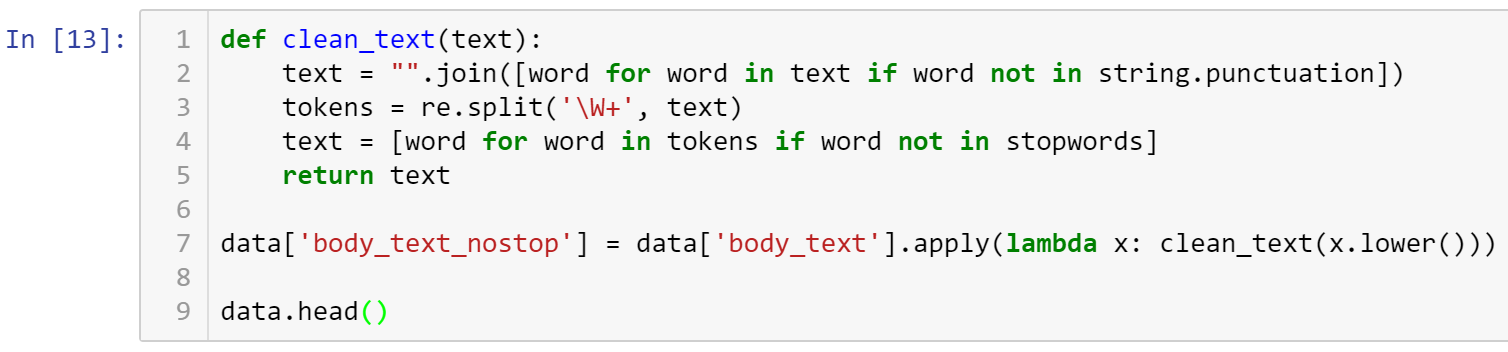


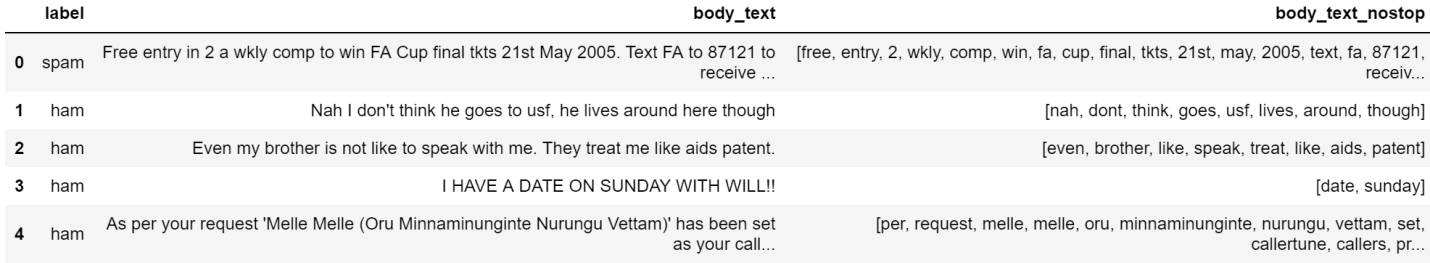
Printing the first 5 rows …



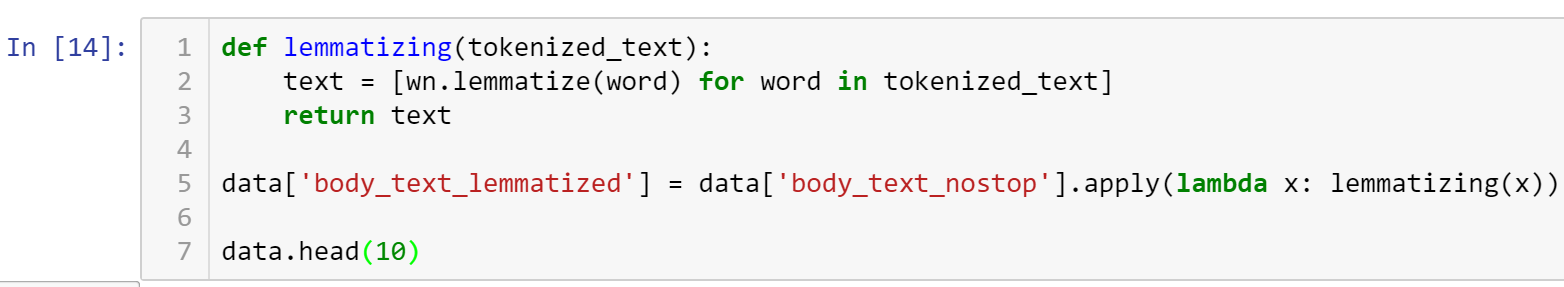


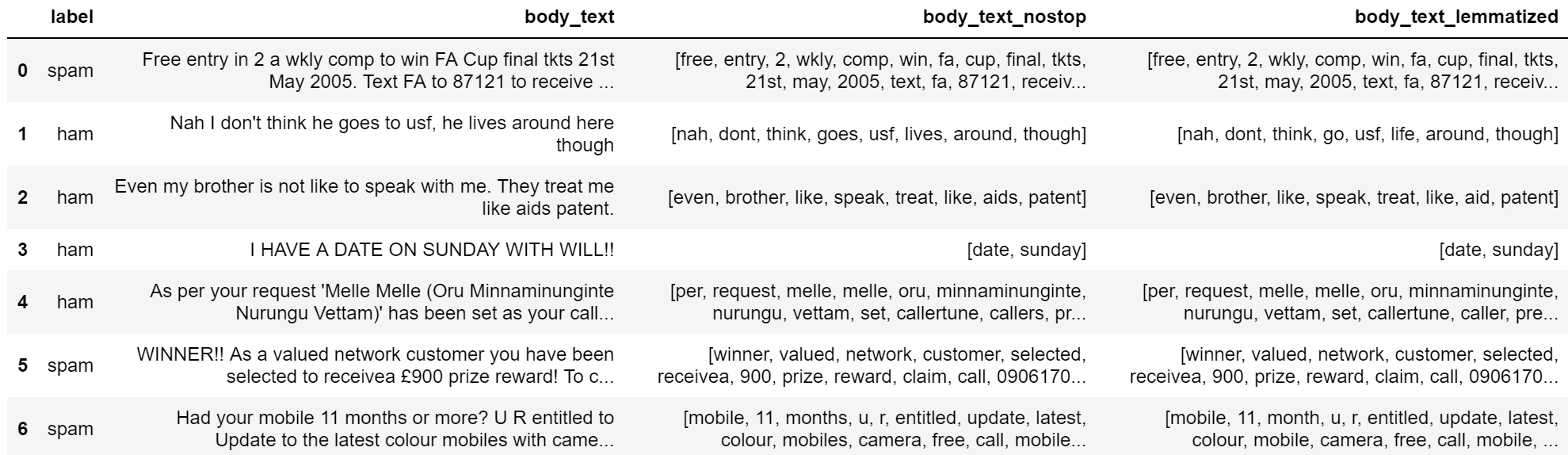
Now, cleaning the data up …





Let’s jump to the lemmatizing portion of this process …





Note in the last two columns how lives translated to life and mobiles into mobile. In other sophisticated datasets such as academia, lemmatizer will add tremendous meaning to your data 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.