TEXT-GATHERING, CLEANING, AND MODELING

I put together the following code as an example of the text gathering, cleaning, and modeling process.

I used Wikipedia as a "toy" case, however this workflow could be adapted for any API with search functionality (e.g. Twitter).

The code is broken into three notebooks

- 01-wiki-crawl.ipynb (Gather+clean+label text data)
    - This code crawls through wikipedia to get a bunch of text data
    - The code lets the user specify search category topics.
        - The more different the topics are, the easier the classification will be.
        - For example, i used (pizza, metallurgy, basketball)
    - It then searches wikipedia for articles related to these topics
    - Loops over the wikipedia pages and gets the text from the wikipedia pages
    - Breaks the text into chunks (based on a user input specifying the number of sentences per chunk)
    - Each chunk is cleaned and tagged with a "label" (classification) and a numeric "sentiment score" (regression)
    - These cleaned chunks form a corpus of strings with associated tags
- 02-explore-results.ipynb (EDA)
    - This notebook vectorizes the corpus and does some basic NLP exploratory data analysis
- 03-classification-and-regression.ipynb (classification and regression modeling)
    - This notebook vectorizes the corpus
    - The data ends up in the following format
        - X strings from wikipedia articles about different topics (need to be vectorized first)
        - y1 = CATEGORICAL topics (0,1,2)= (pizza, metallurgy, basketball) --> Classification targets
        - y2 = NUMERIC CONTINUOUS (sentiment score -1 to 1) --> Regression targets
    - Partitions data into training and test
    - Then trains the following models
        - Multi-nomial Naive Bayes classifier
        - KNN classifier
            - (with a hyper-parameter tuning demonstrating)
        - KNN regression
            - (with a hyper-parameter tuning demonstrating)