

Final Project - Word Cloud

For this project, you'll create a "word cloud" from a text by writing a script. This script needs to process the text, remove punctuation, ignore case and words that do not contain all alphabets, count the frequencies, and ignore uninteresting or irrelevant words. A dictionary is the output of the `calculate_frequencies` function. The `wordcloud` module will then generate the image from your dictionary.

For the input text of your script, you will need to provide a file that contains text only. For the text itself, you can copy and paste the contents of a website you like. Or you can use a site like Project Gutenberg to find books that are available online. You could see what word clouds you can get from famous books, like a Shakespeare play or a novel by Jane Austen. Save this as a .txt file somewhere on your computer.

Now you will need to upload your input file here so that your script will be able to process it. To do the upload, you will need an uploader widget. Run the following cell to perform all the installs and imports for your word cloud script and uploader widget. It may take a minute for all of this to run and there will be a lot of output messages. But, be patient. Once you get the following final line of output, the code is done executing. Then you can continue on with the rest of the instructions for this notebook.

Enabling notebook extension fileupload/extension...
- Validating: OK

```
In [14]: # Here are all the installs and imports you will need for your word cloud script and uploader widget
import wordcloud
import numpy as np
from matplotlib import pyplot as plt
from IPython.display import display
import fileupload
import io
import sys

Requirement already satisfied: ipywidgets in /opt/conda/lib/python3.6/site-packages (7.4.2)
Requirement already satisfied: ipython>=4.0.0; python_version >= "3.3" in /opt/conda/lib/python3.6/site-packages (from ipywidgets) (7.4.0)
Requirement already satisfied: nbformat>=4.2.0 in /opt/conda/lib/python3.6/site-packages (from ipywidgets) (4.4.0)
Requirement already satisfied: traitlets>=4.3.1 in /opt/conda/lib/python3.6/site-packages (from ipywidgets) (4.3.2)
Requirement already satisfied: ipykernel>=4.5.1 in /opt/conda/lib/python3.6/site-packages (from ipywidgets) (5.1.0)
Requirement already satisfied: widgetsnbextension~=3.4.0 in /opt/conda/lib/python3.6/site-packages (from ipywidgets) (3.4.2)
Requirement already satisfied: setuptools>=18.5 in /opt/conda/lib/python3.6/site-packages (from ipython>=4.0.0; python_version >= "3.3"->ipywidgets) (40.8.0)
Requirement already satisfied: jedi>=0.10 in /opt/conda/lib/python3.6/site-packages (from ipython>=4.0.0; python_version >= "3.3"->ipywidgets) (0.13.3)
Requirement already satisfied: decorator in /opt/conda/lib/python3.6/site-packages (from ipython>=4.0.0; python_version >= "3.3"->ipywidgets) (4.3.2)
Requirement already satisfied: pickleshare in /opt/conda/lib/python3.6/site-packages (from ipython>=4.0.0; python_version >= "3.3"->ipywidgets) (0.7.5)
Requirement already satisfied: prompt_toolkit<2.1.0,>=2.0.0 in /opt/conda/lib/python3.6/site-packages (from ipython>=4.0.0; python_version >= "3.3"->ipywidgets) (2.0.9)
Requirement already satisfied: pygments in /opt/conda/lib/python3.6/site-packages (from ipython>=4.0.0; python_version >= "3.3"->ipywidgets) (2.3.1)
```

Whew! That was a lot. All of the installs and imports for your word cloud script and uploader widget have been completed.

IMPORTANT! If this was your first time running the above cell containing the installs and imports, you will need save this notebook now. Then under the File menu above, select Close and Halt. When the notebook has completely shut down, reopen it. This is the only way the necessary changes will take affect.

To upload your text file, run the following cell that contains all the code for a custom uploader widget. Once you run this cell, a "Browse" button should appear below it. Click this button and navigate the window to locate your saved text file.

```
In [34]: # This is the uploader widget

def _upload():
    _upload_widget = fileupload.FileUploadWidget()

    def _cb(change):
        global file_contents
        decoded = io.StringIO(change['owner'].data.decode('utf-8'))
        filename = change['owner'].filename
        print('Uploaded `{}` ({:.2f} kB)'.format(
            filename, len(decoded.read()) / 2 **10))
        file_contents = decoded.getvalue()

    _upload_widget.observe(_cb, names='data')
    display(_upload_widget)

_upload()

Browse
```

Uploaded `test.txt` (10.63 kB)

The uploader widget saved the contents of your uploaded file into a string object named `file_contents` that your word cloud script can process. This was a lot of preliminary work, but you are now ready to begin your script.

Write a function in the cell below that iterates through the words in `file_contents`, removes punctuation, and counts the frequency of each word. Oh, and be sure to make it ignore word case, words that do not contain all alphabets and boring words like "and" or "the". Then use it in the `generate_from_frequencies` function to generate your very own word cloud!

Hint: Try storing the results of your iteration in a dictionary before passing them into wordcloud via the `generate_from_frequencies` function.

```
In [37]: def calculate_frequencies(file_contents):
# Here is a list of punctuations and uninteresting words you can use to process your text
punctuations = '''!()-[]{};:'"\,.<>./?@#%&*~'''
uninteresting_words = ["the", "a", "to", "if", "is", "it", "of", "and", "or", "an", "as", "i", "me", "my", \
"we", "our", "ours", "you", "your", "yours", "he", "she", "him", "his", "her", "hers", "its", "they", "them", \
"their", "what", "which", "who", "whom", "this", "that", "am", "are", "was", "were", "be", "been", "being", \
```

```

# LEARNER CODE START HERE
non_punctuation_text=""
for char in file_contents:
    if char not in punctuations:
        non_punctuation_text+=non_punctuation_text+char
words=non_punctuation_text.split()
clean_words=[]
frequencies={}
for word in words:
    if word.isalpha():
        if word not in uninteresting_words:
            clean_words.append(word)
for alpha_word in clean_words:
    if alpha_word not in frequencies:
        frequencies[alpha_word]=1
    else:
        frequencies[alpha_word]+=1

#wordcloud
cloud = wordcloud.WordCloud()
cloud.generate_from_frequencies(frequencies)
return cloud.to_array()

```

```
In [38]: # Display your wordcloud image
```

[illegible]

In []: