Metis Project 4 - Book Recommendation

Gregory Lull

Design:

The focus of this project was the create a book recommendation system based on text analysis as opposed to book ratings. My idea was that readers could be recommended books based on vocabulary and grammatical structures so that a recommended book would be similar enough that it's easy to read and worth the time dedication.

The dataset I used was my personal and friend's collection of Kindle books, and Project Gutenberg's free books. These were converted into plain text files using Calibre 4.2.0, cleaned and tokenized with Spacy and TF-IDF (bigram), dimension reduced with LSA (n=50), clustered with KMeans (k=20), and then sorted with cosine similarity. The final result is that a given book would go through the same pipeline and a top 10 list would be returned to the user.

The results were a little bit mixed. I intentionally set aside a "test" set which were books in a series in the beginning to measure how this recommendation system performed at the end. For example, for the series Harry Potter, Dune, Ender's Game, Game of Thrones, Narnia, Hitchhiker's Guide, I took 1-2 books from each series and set it aside. What I expected was that given a book from a series, the recommendation system should return that series first and foremost, and then some other books that are similar.

For something like Harry Potter and Shakespeare, their clusters and returned books only had books from that series and that author. But for books like Dune, Narnia, Ender's Game the returned books actually didn't contain all the series and had some other books. One neat result was that Jane Austen's Pride and Prejudice returned a list where 7/10 of the books were set between 19th to 20th century, and were set in the UK.

Data Collection:

MVP

For the MVP a collection of Kindle books I own and that my friends own. These books are generally more recent, and contains both fiction and non-fiction by authors such as such as JK Rowling, Stephen Hawking, and Sir Arthur Conan Doyle.

1) Processing: These files were in the Kindle .mobi format and I needed to convert it to a text format so that the python libraries could read it. To do so I used Calibre 4.2.0 and its CLI to batch convert .mobi files to .txt files, this was accomplished via a python file that could execute shell commands. The conversion takes about 26 minutes for 550 files (400mb), which could probably be parallelized, e.g. convert files that start A-C in one shell, and then D-F in another shell, and so on.
2) Note: I think the .mobi files have metadata I can extract such as ISBN, genre, publisher, etc., but this was out of scope for the MVP. These options are listed in the Calibre API docs https://manual.calibre-ebook.com/generated/en/ebook-convert.html.

Beyond MVP

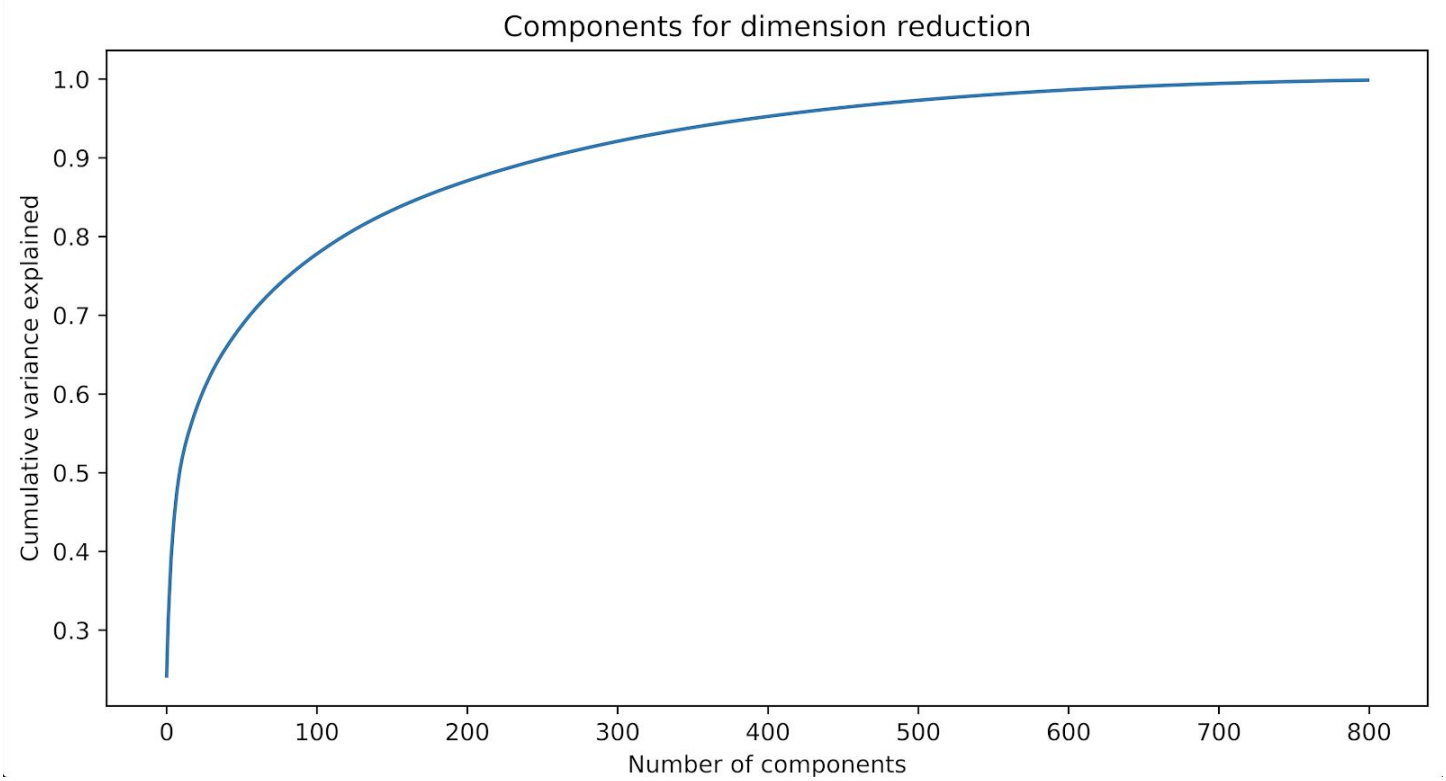
Beyond the MVP I downloaded the whole Project Gutenberg library up to Oct 2018 from https://wiki.kiwix.org/wiki/Content. The file is 50GB and is a compressed .zim format which then required uncompressing to extract each individual book that I needed.

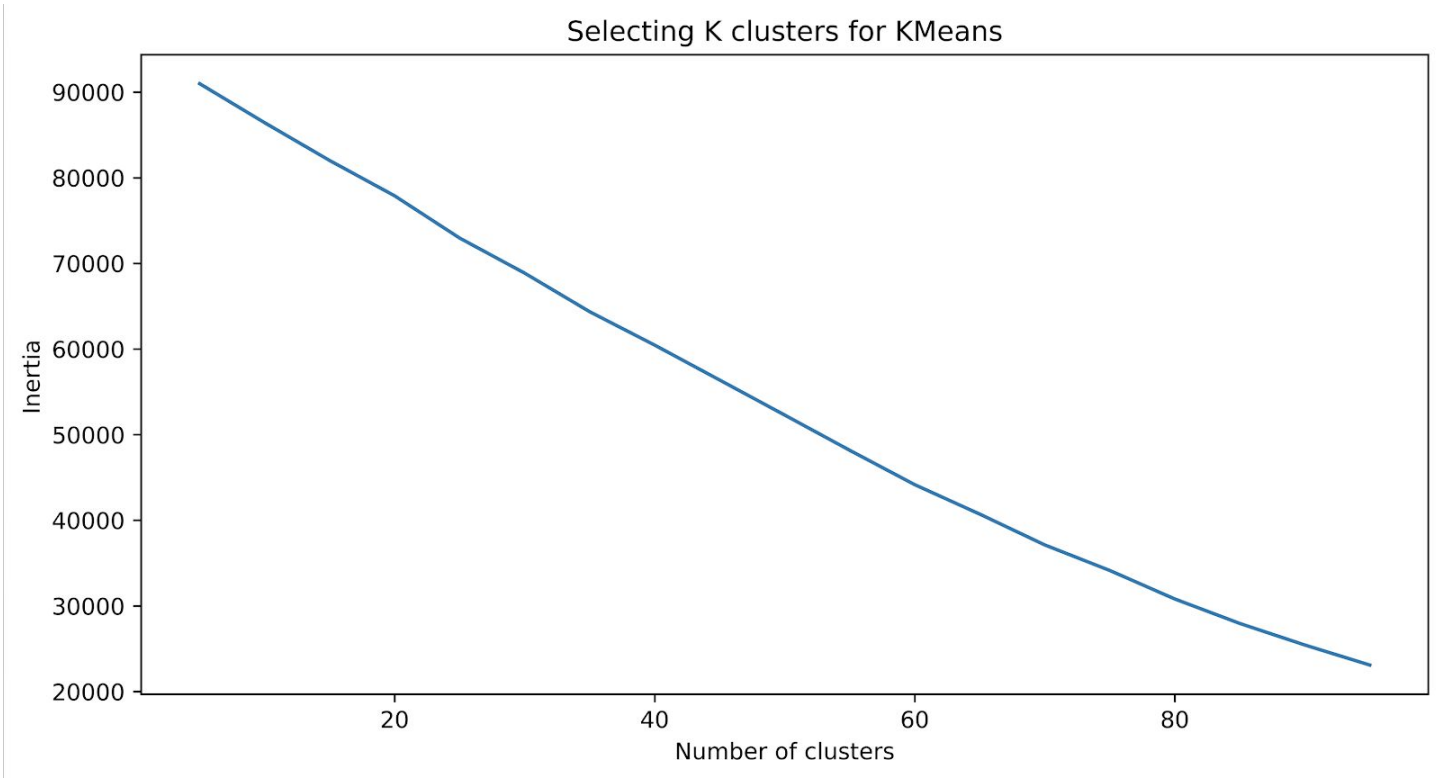
3) The .zim file is apparently very easily viewed if I used zim reader, such as Kiwix's macOS desktop app, however getting the individual files was much more difficult and required compiling an open source library called libzim https://github.com/openzim/libzim. The directions themselves wasn't working properly for my systems (macOS laptop nor Ubuntu desktop) and I had to spend a couple hours online troubleshooting and installing/reinstalling dependencies in the correct order.
    a) Advice on package requirements and troubleshooting an issue with <include unicode> https://stackoverflow.com/questions/33259191/installing-libicu-dev-on-mac
4) Directions on how to use zim-tools to "unpack" the single 50gb gutenberg.zim file into individual files and pictures. My laptop ran out of space and was only able to mostly unpack the HTML versions and not the epub version.
    a) Seeing this answer is what led to compiling and installing the zimlib on my laptop (attempted for Ubuntu desktop as well but could not get it to work). https://superuser.com/questions/970857/how-to-uncompress-a-zim-file
5) These are alternate directions for downloading all of the Gutenberg library files one by one. https://webapps.stackexchange.com/questions/12311/how-to-download-all-english-books-from-gutenberg
6) This is a list of Gutenberg zim files for different languages. I downloaded a smaller zim file ~10mb to test the zim commands first before executing on the 50gb file. https://download.kiwix.org/zim/gutenberg/

Modeling and Analysis:

I didn't have a metric to use for this. Originally I was thinking I could create my own metric, such as checking if books from the same series were in the same cluster, or if they were separated into multiple clusters. But I was worried that this would "overfit" my data, and cluster in ways that was unexpected.

For the modeling and validation I cleaned and lemmatized the text with Spacy, vectorized with a combination of CountVectorizer or TFIDF with bigrams, tried LSA and NMF to reduce the dimensions, and then tried KMeans and DBSCAN for the clustering. The final results for each cluster was then sorted with cosine similarity. I picked my LSA components by graphing a cumulative variation ratio chart, and it seemed like 50-100 components capture about 50-80% of my variation. For KMeans I plotted an inertia chart, but it didn't seem like there was an elbow anywhere, so I just set it at k=20, and also k=10 to see the difference (not much of a difference).





Conclusion:

I definitely felt humbled and an appreciation for NLP, this was extremely hard, and my computer actually ran out of space, so I think I would like to learn how to use cloud computing to run my models next time. Some lessons learned if I try this again would be the initial cleaning. I only used bigrams and lemmatization, but some of my results I think were hijacked because of the pronouns and nouns that were specific to books. For

example during topic modeling, the 3rd component (10% variation) had a lot of Harry Potter specific words, and although it makes sense that Harry Potter should group with its own books, I would've liked to remove/replace those types of words and see how the sentences/grammar cluster together.

Some other things I would've liked to do is use user ratings to create recommendations, more books (I only had 1000 books), and to create a webapp, but I ran out of time.

Miscellaneous:

Visual Studio Code environment:

For the previous project I was having issues importing my own functions in subfolders, this prevented me from properly running my code / debugging and I was not able to find a solution at the time.

This project I made progress by following the directions in this issues thread https://github.com/Microsoft/vscode-python/issues/3840, and vscode docs https://code.visualstudio.com/docs/python/environments#_use-of-the-pythonpath-variable.

My folder structure:

```
project_4_folder/
    src/
        utilities/
            doc_utils.py
            Token_utils.py
    main.py
```

In my main.py i am trying to import my own code:

```
from src.utilities import doc_utils as dutils
```

Step 1: In vscode workspace settings (cmd + shift + p, workspace), search terminal.integrated.env, edit the settings.json and add this line to any existing settings.

```
{
    "python.envFile": "${workspaceFolder}/.env"
}
```

Step 2 Create a ".env" file in the project_4_folder and inside the file set the path to your folders that contain your code.

```
PYTHONPATH=/Users/greg/metis/nlp-classifier/src:${PYTHONPATH}
```

Step 3 Create a ".profile" file and export that same path so I can run my script from the command line (execute `source .profile` once per shell)

```
export PYTHONPATH="/Users/greg/metis/nlp-classifier/src:$PYTHONPATH"
```