Colin Savage LING 401.001

I looked at the uniqueness and identifiability of J.R.R. Tolkien compared to other authors of his time and style (specifically, C.S. Lewis, P.L. Travers, and Richard Matheson). I compared five aspects: vocabulary richness (the number of unique words divided by the total number of words), average sentence length, the 50 most common words (stopwords removed), the 50 most common stopwords, and the 25 most common punctuation marks.

I wanted to look at this problem because, having read Tolkien's works many times, I feel that his writing style is very unique and identifiable, and I wanted to see if it actually is. I also wanted to compare major aspects of linguistic analysis and see how distinct they are among different authors.

To get this data, I took *The Hobbit* and the three books of *The Lord of the Rings* trilogy by Tolkien, *The Lion, the Witch, and the Wardrobe* and *Prince Caspian* by Lewis, *Mary Poppins* by Travers, and *I Am Legend* by Matheson. I split each book into each of its chapters, each chapter in a different txt file, each book organized into a folder, each author's works organized into a folder together. I also took a small sample of chapters from each author and created a test folder in the same format as a regular book so that I could test the identifiability of each author. The full books can be found online for free, most commonly as pdfs. I just copied and pasted the words into txt files.

I designed this program on the fly, with a general idea of what I wanted but never with really clear intent. Despite that, it seems to work relatively well! I started by creating data frames for each author with data columns for book title, chapter, and raw, uncleaned text. Next, I cleaned the data by making it all lowercase, getting rid of newline characters, removing chapter markers, whitespace, and numbers, and replacing every mention of a name (as determined by a master list of all names in all the books) with an "X". To avoid overwriting data, I just made the clean text as a new column in the data frames. Then I created another column for the text with stopwords and punctuation removed. Then another column for only the stopwords, then another for only the punctuation. Next, I wrote a function (or method, or whatever they're called in Python), to get the ratio of unique words to total words in a book. Then I wrote a function to get the average sentence length, one to get the 50 most common words (with stopwords removed), one to get the 50 most common stopwords, and one to get the 25 most common punctuation markers (because I think punctuation is important in writing style, but it's unrealistic to compare 50 different punctuation markers). Then I created a Book object to represent (surprise!) a book and all the data for that book. Then I created an Author object to represent an author and all the books they wrote, along with the average data for all their books. After putting all the data into the class types I made, I wrote a function to compare the data from two books and return the differences in the five values, then a value that takes the five comparison data points and merges them with equal weight into a single value that represents the similarity between two books between 0 and 1. The last function I wrote was a test function to take in a new text and compare it to each book written by each author, creating a dictionary to represent each author and the similarity of the sample text with everything that author has written (in my data set). Then it prints out the author with the highest similarity to the sample text. Lastly, I passed the four test samples I'd made earlier into the test function to see if they predicted the correct author for the sample. They did!

For ease of comprehension in the Jupyter notebook, I just printed out the results. Here is the output:

```
The author of this text is most likely Tolkien with similarity 75.16 %

The author of this text is most likely Lewis with similarity 70.25 %

The author of this text is most likely Travers with similarity 66.46 %

The author of this text is most likely Matheson with similarity 69.81 %
```

This doesn't show much on its own because I designed it to hide the complexity and just show the results, which I'm realizing as I write this report is maybe not the best for this scenario. However, it is currently 1:21 AM and I have an exam at noon, so I'm just going to have to explain as best I can without figures. I just multiplied the summary similarity value by 100 to get the percent similarity and rounded it to two decimal places so it looks nicer. All the values are factored in to create the single number behind the scenes, but I promise if you look at the code you'll see that it's real.

If I had more time and programming experience, and perhaps better time management skills, I would expand this project by getting more data for each author and more authors, comparing things like cosine similarity, part of speech distribution, and semantics. I would also want to modularize it more to make it easier to add authors and books.

Being the only person working on this project, I did everything.