Project Overview:

I used Project Gutenberg as my data source to copy Niccolo Machiavelli's "The Prince" and "The Discourses". There has been an academic debate over whether Niccolo Machiavelli's "The Prince" and "The Discourses" expressed similar ideas and were just misunderstood. I applied cosine similarity on the texts at different parts of the book. I hope to create a visual representation of the difference of the books as they progress and use the results to lend credence to one side of the debate over the similarity of the books.

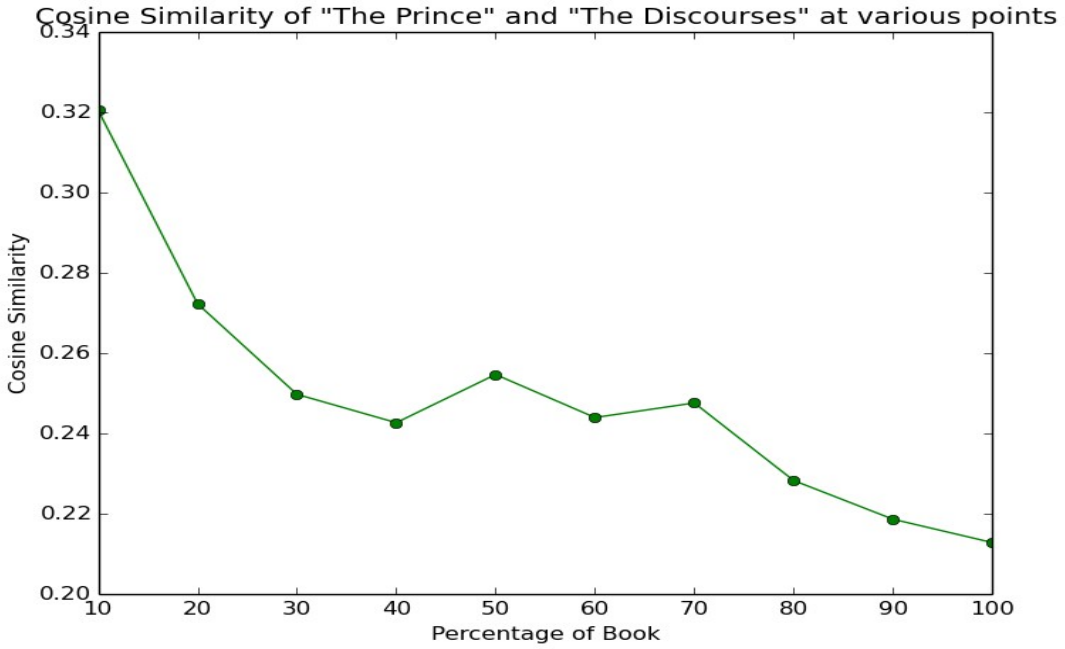
Implementation:

I adapted ThinkPython code to create a histogram of the texts up to a certain percentage of the text. Then using the histograms I applied cosine similarity to the two dictionary histograms, which were my "vectors". As a system there are two parts: creating the vectors and applying cosine similarity analysis.

While designing code I had the option between going through both lists and making sure they had the same keys(add 0 value if it doesn't exist) and ignoring if one entry exists in the other or equals 0. I realized that the other method is more efficient because I wouldn't have to go through both lists again. In terms of cosine similarity analysis, I could just do the second method because if it doesn't exist in one then it will not matter for cosine similarity

Results:

The book's similarity over time was an eventual decline. However, interestingly, there were certain points where it was more similar at that point in the books than at the previous points.



In terms of the real world debate over whether "The Prince" and "The Discourses" which were both written by Niccolo Machiavelli but at different points in his life had a similar message and we were just interpreting it wrong, cosine analysis gave a 0.212744656294 score. This indicates that "The Prince" and "The Discourses" according to cosine similarity are not very similar books, providing support for the side that they are not similar.

Reflection:

I think that the project went well, I was able to support one side of an academic debate. At the same time I got practice with word processing, learned how to plot in python, and observed an interesting effect in the cosine similarity of the book over time. I think my project could've been a bit more complicated, but I got good practice and learned some new things. For unit testing I tested each part with small files or dictionaries to test functionality