28/09/20: Researched AI models, decided that initial concept and testing would be best. Had previously settled on using Python 3.8, which I am familiar with and has a whole host of useful tools for the applications I intend.

5/10/20: Decided to try getting the dataset from open source website Project Gutenberg

12/10/20: Skipped a week, a lot of hw =(

19/10/20: Started coding the required approach using the requests module / learning how to use said module

20/10/20: Learnt about using Beautiful Soup, a useful if strangely named module for processing the data obtained by the requests module

22/10/20: Succeeded in getting the database required, using a mirror website of Project Gutenberg (a free access repository of writing)

2/11/20: Read about Markov chains to get a base understanding before seeing examples of how it is coded in practice

9/11/20: Began using TensorFlow to treat the poetry in my database for later analysis. Currently unused, but learning how perprocessing works was an interesting endeavour

16/11/20: Created a tool that uses the website RhymeZone to find rhymes of words and return the data back to the program, thus avoiding the length process of coding a rhyme finder. Changed the markov chain function to give rhyming lines.

19/11/20: Refined the rhyme finder tool as it took 1.2 seconds per search (as measured with a a tool I had programmed previously)

Currently, requests took 0.6 seconds to get the code of the website, another 0.6 for the rhymes to be isolated and treated. Attempted to get the latter time down by using a better parser called lxml, as well as a character recognition tool called cchardet based in CPython (a faster variant of Python if more difficult to work with). Seeing the effectiveness of CPython, I am inclined to try programming in it at least partially when time is an important factor.

The total time has been reduced to 0.6 seconds average, the processing time has been reduced to an almost negligible level!