# Exercise – Using Hadoop

Zach Adair

12/4/2018

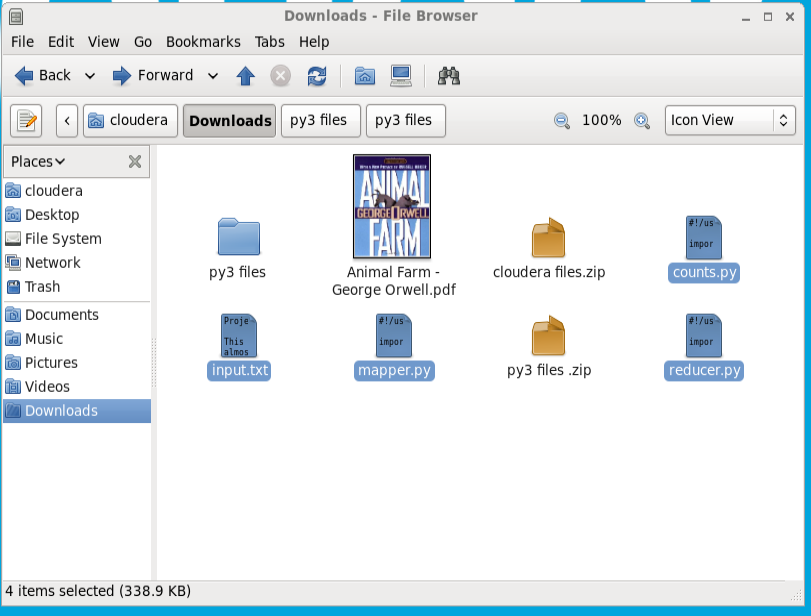
MSDS 600

## Objective

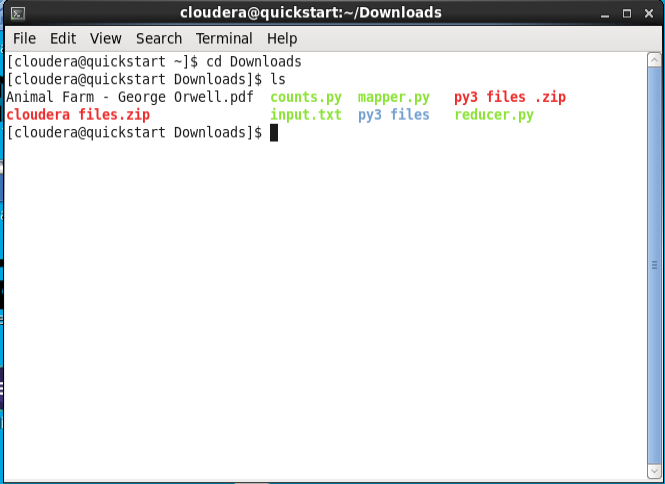
Use Cloudera and Python to run a text analysis, follow the steps outlined in the exercise write-up and take screen shots of the steps taken.

1. Download files into Cloudera.

I was struggling to get my Cloudera machine to connect with my original machine, so I did the shortcut and downloaded them on my Cloudera machine through WorldClass.



1. Next go into the terminal and make sure I have the files in there as well.



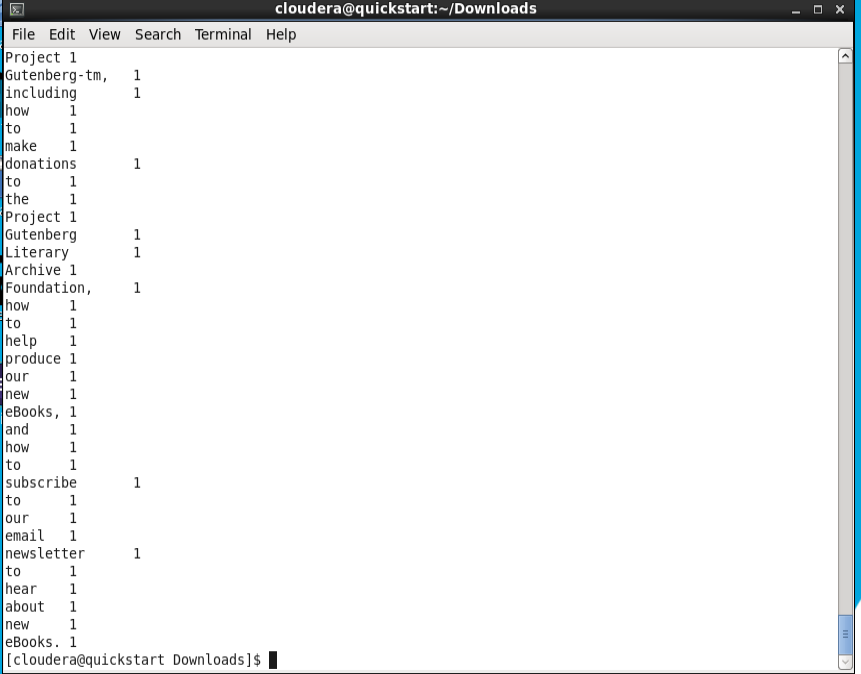
1. Run the counts.py code in Cloudera and check the result from the application.



The numbers next to each letter seem to be a sort of value associated with each word. Also as the dictionary grows in size, the effect that has to have on the run time of this code has to be troublesome because for each word, a value is associated with it, and the memory a computer would have to use to keep all that data would be bad because no matter how much memory you have you would eventually run out of space with the addition of more and more words. The output this code spit out too is so large that it can’t be fit all on one page, hence making it hard to read and analyze effectively.

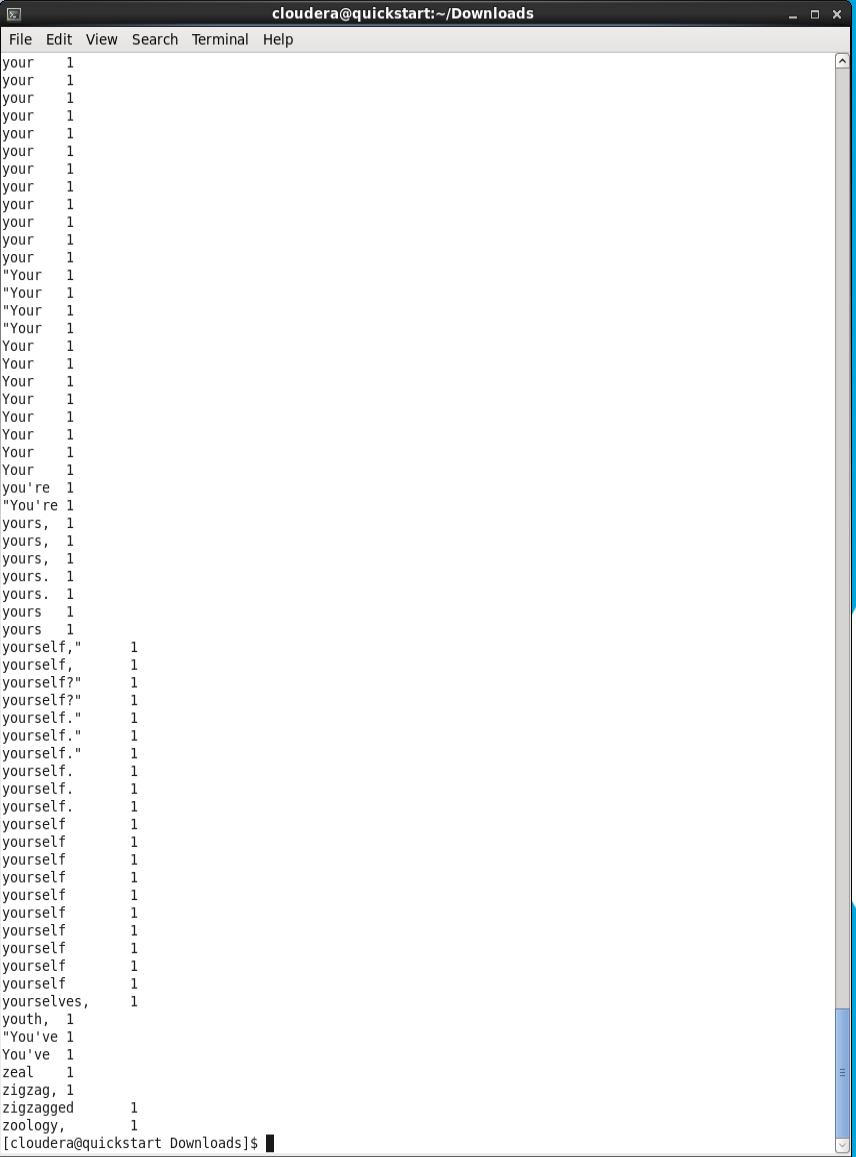
Thankfully, with MapReduce we can solve both these problems.

1. Implement the mapper, running the code ./mapper.py < input.txt



This is a much cleaner view of our text data, one line per word, we still have a problem our output though because it is not sorted and still could be made more readable.

1. Sort the words in alphabetical order using: ./mapper.py < input.txt | sort



Now the output looks much cleaner, it’s sorted and easy to read. Now that the data has been mapped and sorted, the final step is to do Reduce part of MapReduce.

1. Add the reducer to the process by executing the command: ./mapper.py < input.txt | sort | ./reducer.py



With the addition of the reducer to the process, we now see the output is very readable, clean and is an easy way for an analyst to get results and see how much of a specific word is used throughout a given text.

## Reflection

After going through the MapReduce process for the first time it is much easier to understand the steps it goes through and how effective it is for analyzing text data. The code behind the mapper.py and reducer.py were not too complicated but very effective in how it impacted input.txt. In addition to the sorting step, done before the reducer, it allows an analyst to take large text and messy text scripts and create data that can be analyzed quickly and efficiently.