Homework 2

Collaboration Statement

The LaTeXtemplate I am using can be found on *overleaf.com*.

I used the textbook extensively to design my algorithm. I also discussed potential approaches with Jack Frumkes.

1 Algorithm Description

The algorithm is written in Python and follows the pseudocode outlined in the textbook. It can be executed as follows:

```
python3 apriori.py <min_support> <input_file_path> <output_file_path>
```

Where *min_support* is an integer representing the minimum support count. The other two attributes are self-explanatory.

Information on the methods and program structure can be found in the code, which is nicely documented with docstrings and type hints.

The Apriori portion of the program takes around 60 seconds to run on the `T10I4D100K.txt` dataset with support count `500` on my laptop. This seems reasonable.

The program appears to run correctly on other datasets from http://fimi.uantwerpen.be/data when compared to the `Pandas` implementation of Apriori, albeit a bit slower. Lowering the minimum support count seems to slow down the runtime of the algorithm quite significantly.

I made extensive use of sets and dictionaries in order to have O(1) access to support counts. I also cached non-frequent itemsets in order to prune itemsets faster, at the cost of increased space complexity.

This project has helped me better understand the limitations of Apriori and why FP-Growth is generally faster.

2 Code

The code for this program is in a Python file called *apriori.py*. I'm not pasting it in here for readability purposes. The code is nicely documented with Python docstrings and type hints, so you shouldn't have too much trouble navigating it.