Assignment 1 - Part 2

Student: Duncan Ferguson
Student Id: 871641260
Class: Comp 4431-1
Assignment: Assignment 1 part 2
Date: 10/26/2021
Group Members from Assignment 5: Emma Bright, Mike Santoro

Association Rules were originally created for mining information for market baskets. But, they can provide insight into other applications also. In this part of the assignment you will use the mlxtend library functions apriori( ) and association_rules( ) to explore text data. Specifically, we will look at the text from three books:

- Sense and Sensibility, by Jane Austin, published 1811
- A Tale of Two Cities, by Charles Dickens, published 1859
- On The Origin of Species, by Charles Darwin, published 1859 (although written over a 20 year period preceding that date).

All three books were written in english by english people at "roughly" the same time, hence one could assume some commonality in writing style.

Mining using association rules assumes a database of a large number of transactions with some number of items per transaction. I have processed the text of these books to be presented in a similar fashion, where each "transaction" is now a sentence from the book with the words comma-separated. Note, some of the sentences in the books are quite long which would result in intractable run times (and enormous space to hold all possible itemsets). As we saw from our reading, the number of subsets to be considered increases exponentially with the number of items per transaction. For example, assume a sentence has 60 words. Then we might need to consider all possible subsets of size 60, 59, 58, 57, ... and 1. The number of these would be: Choose(60,60) + Choose(60,59) + Choose(60,58) + ...Choose(60,2) + Choose(60,1). Clearly not tractable!

To make this problem tractable I have processed the texts to the they contain at most N words per line (transaction), where N = {10,12,14}. For example, assume N = 14. I processed the file into a list of sentences. Then, for each sentence, if there were more than 14 words I cut into into at most 14 work chunks. The beginning of a new sentence starts a new line. I also removed all punctuation and made everything lower case. As far as text analysis goes, this loses some useful information, but it greatly simplifies the problem and this assignment is more about understanding how to use association rules and frequent itemsets.

On my computer the files for N=14 ran very quickly, about 5 seconds or less depending on the min-support value used, but I have provided the smaller sets just in case you need to use them due to having a slower computer. To figure out which set of files to use, I suggest you run your code on the 10 words per line, 12 words per line, and 14 words per line data sets in turn to see how fast they run. If you computer works well on the 14, only use the 14 for all three texts. If you need to use the 10 or 12, only use that value for all three texts.

You are to explore each of the three texts and see what you can determine about the texts individually, and compared to each other. Write up a summary of commonality and differences explaining the parameter values used. I suggest you look at frequent 1-itemsets, 2-itemsets and possibly 3-itemsets as well as association rules. As a starting point I suggest you start with a min-support value of 0.02 and a min-confidence value of 0.2. Then increase/decrease both as you see fit. Only consider those association rules that have a lift value > someValue, where someValue is around 1.1 .. 1.5.

The same values may not work well for each of the datasets, it is fine to use different values for different texts, just make sure to explain what values you used and whey.

Write up your experiments as a report and upload the report as a .pdf file. Also attach your code for one of your experiments (all the others should be quite similar). You do not need one file that runs everything, how much you "automate" via python scripting is up to you.

30
```
# Adding in all the import statements
from csv import reader
import pandas as pd
from mlxtend.frequent_patterns import apriori, association_rules
from mlxtend.preprocessing import TransactionEncoder

# pd.set_option('display.max_rows', None)
# pd.set_option('display.width', None)
```

31
```
# importing the texts
def fill_book_list(file_name):
    """This function reads in the book"""
    book = []
    with open(file_name, 'r') as read_obj:
        csv_reader = reader(read_obj)
        for row in csv_reader:
            book.append(row)
    return book

book1 = fill_book_list('input_SenseAndSensibility_10.csv')
book2 = fill_book_list('input_TaleOfTwoCities_10.csv')
book3 = fill_book_list('input_OriginOfSpecies_10.csv')
```

Creating a function for using the transaction encoder to transform the books into an 1-hot boolean encoded numpy array and convert into a dataframe for the convenience of passing into apriori

```
32 def transact_1_hot(book):
       """Creating a function to easily run the apriori and edit the min_support, min_threshold, and confide
       te = TransactionEncoder()
       te_ary = te.fit(book).transform(book)
       df = pd.DataFrame(te_ary, columns= te.columns_)
       return df

   book1_df = transact_1_hot(book1)
   book2_df = transact_1_hot(book2)
   book3_df = transact_1_hot(book3)
```

Displaying the head of all three books

```
33 book1_df.head()
```

33

	200	7000l	_	_a	_all	_almost_	_always_	_am_	_amiably	_and	...
0	False	False	False	False	False	False	False	False	False	False	...
1	False	False	False	False	False	False	False	False	False	False	...
2	False	False	False	False	False	False	False	False	False	False	...
3	False	False	False	False	False	False	False	False	False	False	...
4	False	False	False	False	False	False	False	False	False	False	...

5 rows × 9564 columns

```
34 book2_df.head()
```

34

	1757	1767	211792	_	_am_	_are_collected	_at_	_bid_	_do_	_don	...
0	False	False	False	False	False	False	False	False	False	False	...
1	False	False	False	False	False	False	False	False	False	False	...
2	False	False	False	False	False	False	False	False	False	False	...
3	False	False	False	False	False	False	False	False	False	False	...
4	False	False	False	False	False	False	False	False	False	False	...

5 rows × 13184 columns

```
35 book3_df.head()
```

35

	000	1	10	100	1000	101	102	107	107330	107416	...	zealan
0	False	False	False	False	False	False	False	False	False	False	...	False
1	False	False	False	False	False	False	False	False	False	False	...	False

	000	1	10	100	1000	101	102	107	107330	107416	...	zealan
2	False	False	False	False	False	False	False	False	False	False	...	False
3	False	False	False	False	False	False	False	False	False	False	...	False
4	False	False	False	False	False	False	False	False	False	False	...	False

5 rows × 10178 columns

19 ## Creating a function to run the apriori algorithm

36
```
def run_apriori(df, min_support_feed=0.02, min_threshold_feed=0.03):
    """This function runs through the apriori algorithm"""
    freq_items = apriori(df, min_support=min_support_feed, use_colnames=True)
    freq_items['length'] = freq_items['itemsets'].apply(lambda x: len(x))
    freq_items.sort_values(["length", "support"], inplace=True, ascending=True)
    rules = association_rules(freq_items, metric="confidence", min_threshold=min_threshold_feed)
    return rules

book1_rules = run_apriori(book1_df)
book2_rules = run_apriori(book2_df)
book3_rules = run_apriori(book3_df)
```

Displaying the head of all three books

37
```
book1_rules.head()
```

37

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leve
0	(to)	(not)	0.245660	0.080227	0.020403	0.083052	1.035212	0.00
1	(not)	(to)	0.080227	0.245660	0.020403	0.254310	1.035212	0.00
2	(i)	(have)	0.120064	0.053738	0.020472	0.170507	3.172921	0.01
3	(have)	(i)	0.053738	0.120064	0.020472	0.380952	3.172921	0.01
4	(as)	(to)	0.070613	0.245660	0.021163	0.299706	1.220003	0.00

38
```
book2_rules.head()
```

38

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	lev
0	(was)	(to)	0.098604	0.188039	0.020340	0.206281	1.097011	0.00
1	(to)	(was)	0.188039	0.098604	0.020340	0.108169	1.097011	0.00
2	(to)	(his)	0.188039	0.098239	0.020461	0.108815	1.107653	0.00

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	lev
3	(his)	(to)	0.098239	0.188039	0.020461	0.208282	1.107653	0.00
4	(i)	(and)	0.103279	0.264177	0.020583	0.199295	0.754397	-0.0

39
```
book3_rules.head()
```

39

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	lev
0	(that)	(it)	0.113744	0.057124	0.020010	0.175921	3.079657	0.01
1	(it)	(that)	0.057124	0.113744	0.020010	0.350294	3.079657	0.01
2	(the)	(but)	0.440445	0.048181	0.020345	0.046193	0.958744	-0.0
3	(but)	(the)	0.048181	0.440445	0.020345	0.422274	0.958744	-0.0
4	(of)	(be)	0.355765	0.085071	0.020457	0.057502	0.675931	-0.0