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.

858
```
# 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
```

859
```
# 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_14.csv')
book2 = fill_book_list('input_TaleOfTwoCities_14.csv')
book3 = fill_book_list('input_OriginOfSpecies_14.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.

860
```
def transact_1_hot(book):
    """Creating transact 1 hot boolean encoded numpy array """
```

```
    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

861
```
book1_df.head()
```

861

	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 × 9659 columns

862
```
book2_df.head()
```

862

	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 × 13254 columns

863
```
book3_df.head()
```

863

	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
2	False	False	False	False	False	False	False	False	False	False	...	False

	000	1	10	100	1000	101	102	107	107330	107416	...	zealan
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 × 10251 columns

19 Creating a function to run the apriori algorithm

```
def run_apriori(df, min_support_feed=0.02, min_threshold_feed=0.2, lift=1.1):
    """This function runs through the apriori algorithm taking in min support
    min threshold which is confidence and min lift. It returns association rules and frequent items"""
    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(["support", "length"], inplace=True, ascending=False)
    rules = association_rules(freq_items, metric="confidence", min_threshold=min_threshold_feed)
    asso_rules = rules[rules["lift"] >= lift].copy()
    asso_rules.sort_values(["lift"], inplace=True, ascending=False)
    return asso_rules, freq_items

book1_rules, book1_freq_items = run_apriori(book1_df)
book2_rules, book2_freq_items = run_apriori(book2_df)
book3_rules, book3_freq_items = run_apriori(book3_df)
```

Taking a look at frequent 1-itemsets that are both strong and interesting

```
def filter_freq_items(freq_items,items=1):
    """This function filters the frequent items so that we can look a certain n-itemsets"""
    filtered_freq_items = freq_items[freq_items["length"] == items].copy()
    return filtered_freq_items
```

The five most common one words for every book with a min-support of 0.02

```
book1_freq_items_1 = filter_freq_items(book1_freq_items)
book1_freq_items_1.head(5)
```

866

	support	itemsets	length
61	0.300081	(to)	1
56	0.283155	(the)	1
4	0.276672	(and)	1
42	0.269650	(of)	1
22	0.174845	(her)	1

```
book2_freq_items_1 = filter_freq_items(book2_freq_items)
book2_freq_items_1.head(5)
```

867

	support	itemsets	length
44	0.421277	(the)	1
3	0.324901	(and)	1
34	0.259023	(of)	1
50	0.232467	(to)	1
0	0.192829	(a)	1

868
```
book3_freq_items_1 = filter_freq_items(book3_freq_items)
book3_freq_items_1.head(5)
```

868

	support	itemsets	length
50	0.522363	(the)	1
36	0.434390	(of)	1
3	0.283951	(and)	1
25	0.257707	(in)	1
56	0.225992	(to)	1

Frequent 1-itemsets

When we look at the frequent item sets for 1 word items from the three books we notice a lot of similarities. All three books have the following words in the top 5;

- The
- To
- And
- Of

Books two and three has a really high support for the word "the". The rankings of the top 5 words varies but there is nothing that is too surprising here as these words appear in most english sentences.

Now looking into the differences we do find that the support ranking order is different. Listed below are the words that appear in the top 5 for one book but not the others:

- "her" (book1)
- "a" (book2)
- "in" (book3)

The five most common two words itemsets for every book with a min-support of 0.02

869
```
book1_freq_items_1 = filter_freq_items(book1_freq_items, 2)
book1_freq_items_1.head(5)
```

869

	support	itemsets	length
177	0.138651	(of, the)	2
105	0.098316	(and, the)	2
190	0.095165	(to, the)	2
106	0.092284	(to, and)	2
101	0.089943	(and, of)	2

870
```
book2_freq_items_1 = filter_freq_items(book2_freq_items, 2)
book2_freq_items_1.head(5)
```

870

	support	itemsets	length
151	0.171316	(the, of)	2
91	0.163672	(the, and)	2
160	0.109771	(to, the)	2
138	0.105437	(in, the)	2
88	0.092120	(and, of)	2

871
```
book3_freq_items_1 = filter_freq_items(book3_freq_items, 2)
book3_freq_items_1.head(5)
```

871

	support	itemsets	length
162	0.311895	(of, the)	2
135	0.164190	(in, the)	2
88	0.163303	(the, and)	2
181	0.126488	(to, the)	2
84	0.126118	(of, and)	2

Frequent 2-itemsets

Right off the bat we can see some major differences by looking at the support values for the top 5 two itemset words. Book3, Darwin, is using a lot of two word itemsets much more. Book1 is showing a lot less support for using the same two word itemsets. Jane Austin's novel was

probably a bit smoother of a read with more variety. To lay out the similarities in black and white below are the similarities and differences for the top 5.

Similarities

- [Of, The]
- [The, And]
- [The, To]
- [Of, And]

Differences

- [To, And] (Book1)
- [The, In] (Book2) (Book3)

Once again there are four 2-item sets appear in the top 5 for all three books. Almost close to having all 5 being similar

The five most common three word itemsets for every book with a min-support of 0.02

872
```
book1_freq_items_1 = filter_freq_items(book1_freq_items, 3)
book1_freq_items_1.head(5)
```

872

	support	itemsets	length
202	0.046187	(the, and, of)	3
209	0.043756	(of, to, the)	3
205	0.030431	(of, her, the)	3
204	0.029711	(to, and, the)	3
208	0.029621	(of, in, the)	3

873
```
book2_freq_items_1 = filter_freq_items(book2_freq_items, 3)
book2_freq_items_1.head(5)
```

873

	support	itemsets	length
174	0.060757	(the, and, of)	3
179	0.044129	(the, in, of)	3
182	0.040032	(to, the, of)	3
176	0.038455	(to, the, and)	3
173	0.038377	(the, in, and)	3

874
```
book3_freq_items_1 = filter_freq_items(book3_freq_items, 3)
book3_freq_items_1.head(5)
```

874

	support	itemsets	length
193	0.089894	(of, the, and)	3
205	0.088268	(of, in, the)	3
215	0.066238	(of, to, the)	3
192	0.047535	(the, in, and)	3
190	0.046869	(of, a, the)	3

Frequent 3-itemsets

Similarities

- [Of, The, And]
- [Of, The, In]
- [Of, The, To]

Differences

- [Of, The, Her] (Book1)
- [The, In, And] (Book2, Book3)
- [The, To, And] (Book1, Book2)
- [A, The, Of] (Book3)

After looking at all the Frequent 1,2,3 item sets it becomes apparent that all of these words are the same. The reason being is that they are all articles, pronouns, prepositions, conjunctions. These words don't really tell us too much about the text or what is in it. Rather it is just showing us the most common words that are used in almost every english sentence.

Looking into the association rules with a min_support=.2 and min_confidence=.02

For right now I'm just going through and print out the association rules so that we can reconfirm that all of these words are mostly "stop words" that do not add too much understanding to what is inside of the books. These words should have high lift values and confidence as they are used with a high frequency but do not relay much value.

The first look is at the top 5 interesting rules. This is done by looking at the lift values.

875
```
book1_rules.head()
```

875

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	le
162	(am)	(i)	0.022508	0.146394	0.020167	0.896000	6.120462	0.
96	(have)	(i)	0.069506	0.146394	0.027370	0.393782	2.689878	0.
72	(you)	(i)	0.084631	0.146394	0.030701	0.362766	2.478008	0.
71	(i)	(you)	0.146394	0.084631	0.030701	0.209717	2.478008	0.
124	(a, the)	(of)	0.044566	0.269650	0.024579	0.551515	2.045302	0.

876
```
book2_rules.head()
```

876

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	
132	(mr)	(lorry)	0.042396	0.025532	0.023089	0.544610	21.330545	
131	(lorry)	(mr)	0.025532	0.042396	0.023089	0.904321	21.330545	
150	(have)	(i)	0.055004	0.126241	0.021434	0.389685	3.086829	
140	(is)	(it)	0.056659	0.130102	0.022301	0.393602	3.025325	
80	(i)	(you)	0.126241	0.088574	0.031127	0.246567	2.783748	

877
```
book3_rules.head()
```

877

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	le
98	(have)	(been)	0.121535	0.064538	0.033932	0.279197	4.326116	0.
99	(been)	(have)	0.064538	0.121535	0.033932	0.525773	4.326116	0.
142	(is)	(it)	0.098100	0.074444	0.027870	0.284099	3.816299	0.
141	(it)	(is)	0.074444	0.098100	0.027870	0.374379	3.816299	0.
119	(that)	(it)	0.147557	0.074444	0.030458	0.206413	2.772737	0.

The first look at the association rules is filter to have the lift values as the highest. The first book contains a lot of words [am, have, you] as antecedents to the consequents I.

The second book [mr] followed by [lorry] makes sense as mr is always going to be before lorry, for Mr Lory.

The third book definitely shows a few different rules. With [have] followed by [been] or [it],[is] would make sense for Darwin describing different types of animals and evolution. This can also be shown by less person pronouns being used.

Filtering rules by support to add an extra look

```
878 book1_rules.sort_values(by=["support"], ascending=False).head()
```

878

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leve
0	(of)	(the)	0.269650	0.283155	0.138651	0.514190	1.815934	0.06
1	(the)	(of)	0.283155	0.269650	0.138651	0.489666	1.815934	0.06
2	(and)	(the)	0.276672	0.283155	0.098316	0.355353	1.254978	0.01
3	(the)	(and)	0.283155	0.276672	0.098316	0.347218	1.254978	0.01
5	(the)	(to)	0.283155	0.300081	0.095165	0.336089	1.119994	0.01

```
879 book2_rules.sort_values(by=["support"], ascending=False).head()
```

879

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leve
1	(of)	(the)	0.259023	0.421277	0.171316	0.661393	1.569974	0.06
0	(the)	(of)	0.421277	0.259023	0.171316	0.406659	1.569974	0.06
3	(and)	(the)	0.324901	0.421277	0.163672	0.503759	1.195793	0.02
2	(the)	(and)	0.421277	0.324901	0.163672	0.388515	1.195793	0.02
4	(to)	(the)	0.232467	0.421277	0.109771	0.472203	1.120887	0.01

```
880 book3_rules.sort_values(by=["support"], ascending=False).head()
```

880

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leve
0	(of)	(the)	0.434390	0.522363	0.311895	0.718005	1.374534	0.08
1	(the)	(of)	0.522363	0.434390	0.311895	0.597085	1.374534	0.08
3	(the)	(in)	0.522363	0.257707	0.164190	0.314322	1.219689	0.02
2	(in)	(the)	0.257707	0.522363	0.164190	0.637120	1.219689	0.02
4	(the)	(and)	0.522363	0.283951	0.163303	0.312624	1.100980	0.01

When sorting the association rules by strongest support we see the same results as looking at the 1-itemset from the analysis above.

Next is a quick look at looking at the largest confidence to look at the strongest rules.

881
```
book1_rules.sort_values(by=["confidence"], ascending=False).head()
```

881

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	le
162	(am)	(i)	0.022508	0.146394	0.020167	0.896000	6.120462	0.
112	(was, of)	(the)	0.045287	0.283155	0.025389	0.560636	1.979964	0.
85	(in, of)	(the)	0.053120	0.283155	0.029621	0.557627	1.969337	0.
124	(a, the)	(of)	0.044566	0.269650	0.024579	0.551515	2.045302	0.
76	(her, the)	(of)	0.056901	0.269650	0.030431	0.534810	1.983351	0.

882
```
book2_rules.sort_values(by=["confidence"], ascending=False).head()
```

882

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	
131	(lorry)	(mr)	0.025532	0.042396	0.023089	0.904321	21.330545	
42	(in, of)	(the)	0.059968	0.421277	0.044129	0.735874	1.746771	
100	(of, was)	(the)	0.038534	0.421277	0.027187	0.705521	1.674723	
52	(to, of)	(the)	0.058077	0.421277	0.040032	0.689281	1.636172	
1	(of)	(the)	0.259023	0.421277	0.171316	0.661393	1.569974	

883
```
book3_rules.sort_values(by=["confidence"], ascending=False).head()
```

883

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	le
149	(same, of)	(the)	0.027722	0.522363	0.027057	0.976000	1.868434	0.
56	(same)	(the)	0.049604	0.522363	0.046426	0.935917	1.791699	0.
71	(on, of)	(the)	0.051748	0.522363	0.040807	0.788571	1.509624	0.
19	(in, of)	(the)	0.113994	0.522363	0.088268	0.774319	1.482340	0.
106	(is, of)	(the)	0.041990	0.522363	0.032010	0.762324	1.459377	0.

When looking at the largest confidence sets we discover some differences between the words that are being used. There is a much higher confidence of [am] followed by [i] for the first book with a really interesting Lift value of 6. [Mr] followed by [Lorry] should have the largest confidence value. Which it does with the highest lift out of all the rules. The most suprising is Darwins uses in the third book. where [Same,Of] followed by [The] has a large confidence value.

All of the looks at the association rules above show strong confidence values, and have interesting rules shown by the lift values being larger than 1. When filtering by lift values for the look at the interesting rules we see a lot of pronoun and article usage for the first two books. Lots of antecedents being followed by the consequents [The].

Removing the Stop words.

After looking at the frequent items and the association rules above it has become apparent that not too much can be taken away from the strong and interesting rules. For the most part, all of the results have given us strong relations between stop words. Theses are the words that are used most in the english language. It is for this reason that I have decided to remove all the Stop words to try and see if we can develop some more interesting rules that give us more meaning to the association rules. After giving this the first run, I have decided to add [mrs, mr, sir, miss] to the stop list words. [S,T] is getting taken out because all punctuation has been removed ['] and show's a high correlation when conducting analysis.

884
```
# Importing the ENGLISH_STOP_WORDS
from sklearn.feature_extraction.text import ENGLISH_STOP_WORDS

def remove_stop_words(book):
    """This function goes through and removes the stop words from the list"""
    additional_stop_words = ['mrs','mr','s','t','sir','miss']  # Adding additional words to remove from t
    stop_words = ENGLISH_STOP_WORDS.union(additional_stop_words)
    new_book = []
    for line in book:
        new_book.append([word for word in line if word not in stop_words])
    return new_book
```

885
```
# Removing all the stop words from the books
new_book1 = remove_stop_words(book1)
new_book2 = remove_stop_words(book2)
new_book3 = remove_stop_words(book3)
```

886
```
# Sending off the new_books to get 1 hot boolean for setup on apriori
new_book1_df = transact_1_hot(new_book1)
new_book2_df = transact_1_hot(new_book2)
new_book3_df = transact_1_hot(new_book3)
```

887
```
# Sending off new books to look at the association rules and frequent items now that the stop words have
new_book1_rules, new_book1_freq_items = run_apriori(new_book1_df)
```

```
new_book2_rules, new_book2_freq_items = run_apriori(new_book2_df)
new_book3_rules, new_book3_freq_items = run_apriori(new_book3_df)
```

1-itemset Frequent Words (With Stop words removed)

The five most common one words for every book now that stop words have been removed. With min_support at .02 and min_confidence at .2

888
```
new_book1_freq_items_1 = filter_freq_items(new_book1_freq_items)
new_book1_freq_items_1.head()
```

888

	support	itemsets	length
1	0.057801	(elinor)	1
2	0.046457	(marianne)	1
4	0.032862	(said)	1
5	0.021878	(sister)	1
0	0.020978	(edward)	1

889
```
new_book2_freq_items_1 = filter_freq_items(new_book2_freq_items)
new_book2_freq_items_1.head()
```

889

	support	itemsets	length
2	0.047203	(said)	1
0	0.025532	(lorry)	1
1	0.021198	(man)	1

890
```
new_book3_freq_items_1 = filter_freq_items(new_book3_freq_items)
new_book3_freq_items_1.head()
```

890

	support	itemsets	length
6	0.094773	(species)	1
7	0.027722	(varieties)	1
2	0.026022	(forms)	1
5	0.024691	(selection)	1
3	0.023287	(natural)	1

Now that all of the Stop words have been removed we can see significant changes to the 1-itemset frequent words. The only word that is similar is the word [said] between book1 and book2. Darwin however speaks a lot more with scientific words. There are only three words that are above the .02 level of support for book2. Book1 and Book2 are focused more around the difference characters "saying" different words.

2-itemset Frequent Words (With Stop words removed)

891
```
new_book1_freq_items_2 = filter_freq_items(new_book1_freq_items,2)
new_book1_freq_items_2.head()
```

891

	support	itemsets	length

892
```
new_book2_freq_items_2 = filter_freq_items(new_book2_freq_items,2)
new_book2_freq_items_2.head()
```

892

	support	itemsets	length

893
```
new_book3_freq_items_2 = filter_freq_items(new_book3_freq_items,2)
new_book3_freq_items_2.head()
```

893

	support	itemsets	length

There are two 2-itemset Frequent Words that appear when the min-support is set at 0.02 and min-confidence is at .2. To look into this further I will set the min_support to .01 and min_confidence at .1 and see if any frequent itemsets appear.

894
```
# Changing the min_support to 0.01 and min_confidence to .1
new_book1_rules_min_01, new_book1_freq_items_min_01 = run_apriori(new_book1_df,
                                                                  min_support_feed=.01,
                                                                  min_threshold_feed=.1)
new_book2_rules_min_01, new_book2_freq_items_min_01 = run_apriori(new_book2_df,
                                                                  min_support_feed=.01,
                                                                  min_threshold_feed=.1)
new_book3_rules_min_01, new_book3_freq_items_min_01 = run_apriori(new_book3_df,
                                                                  min_support_feed=.01,
                                                                  min_threshold_feed=.1)
```

895
```
new_book1_freq_items_2_min_01 = filter_freq_items(new_book1_freq_items_min_01,2)
new_book1_freq_items_2_min_01.head()
```

895

	support	itemsets	length
31	0.010444	(colonel, brandon)	2

```
new_book2_freq_items_2_min_01 = filter_freq_items(new_book2_freq_items_min_01,2)
```

```
896 new_book2_freq_items_2_min_01.head()
```

896

	support	itemsets	length

```
897 new_book3_freq_items_2_min_01 = filter_freq_items(new_book3_freq_items_min_01,2)
    new_book3_freq_items_2_min_01.head()
```

897

	support	itemsets	length
42	0.01619	(selection, natural)	2

When lowering the min_support book1 and book2 now show some more frequent 2-itemsets. Naturally these have lower support, but the frequent occurrences make sense. Book3 with natural selection and colonel brandon in book1. Because we notice that we loose the support here for 2-itemsets there really is no need to explore further into the association rules.

Association Rules (With Stop words removed)

Now we will look first at the rules that are associated with a min_support=.02 and min_confidence=.2 and a lift above 1.1

```
898 new_book1_rules
```

898

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage

```
899 new_book2_rules
```

899

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage

```
900 new_book3_rules
```

900

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage

This first look gives us no rules that are particularly strong or interesting to investigate. It is for this reason that we will start going through and tinkering with the input values to see if we can find some more rules. Starting with the min_support at .01 and min_confidence at .1

```
901 new_book1_rules_min_01
```

901

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	lev
0	(colonel)	(brandon)	0.014855	0.011614	0.010444	0.703030	60.531454	0.0
1	(brandon)	(colonel)	0.011614	0.014855	0.010444	0.899225	60.531454	0.0

902
```
new_book2_rules_min_01
```

902

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage

903
```
new_book3_rules_min_01
```

903

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	lev
1	(natural)	(selection)	0.023287	0.024691	0.01619	0.695238	28.157143	0.0
0	(selection)	(natural)	0.024691	0.023287	0.01619	0.655689	28.157143	0.0

The lift values show some very interesting rules. In book1 brandon is clearly a colonel with a huge lift value of over 60. And Darwin in book3 uses natural selection together with a huge lift value of 28

Finding Association rules that have a lift of above 1.1 by lowering min_support to .005

Because we are not finding many rules at min_support of .01 and min_confidence back at .2 we will lower the min_support to 0.005. Because theses are large books this should help us find more words that appear together and find more association rules. These words may not appear in every sentence. If this were a data sheet that was looking at items purchased together, it would be a bit different. Since they are books with sentences that vary lowering support makes the most sense. The Confidence will show us what words are used together. Lowering this value will not do us any good in finding strong rules.

904
```
# Setting the min_support to .005
new_book1_rules_min_005, new_book1_freq_items_min_005 = run_apriori(new_book1_df,
                                                                    min_support_feed=.005,
                                                                    min_threshold_feed=.2)
new_book2_rules_min_005, new_book2_freq_items_min_005 = run_apriori(new_book2_df,
                                                                    min_support_feed=.005,
                                                                    min_threshold_feed=.2)
new_book3_rules_min_005, new_book3_freq_items_min_005 = run_apriori(new_book3_df,
```

```
                                                    min_support_feed=.005,
                                                    min_threshold_feed=.2)
```

905 `new_book1_rules_min_005`

905

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	lev
2	(lady)	(middleton)	0.012064	0.008643	0.007473	0.619403	71.663635	0.0
3	(middleton)	(lady)	0.008643	0.012064	0.007473	0.864583	71.663635	0.0
0	(colonel)	(brandon)	0.014855	0.011614	0.010444	0.703030	60.531454	0.0
1	(brandon)	(colonel)	0.011614	0.014855	0.010444	0.899225	60.531454	0.0
4	(said)	(elinor)	0.032862	0.057801	0.007113	0.216438	3.744518	0.0

906 `new_book2_rules_min_005`

906

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	lev
0	(defarge)	(madame)	0.019307	0.013160	0.008353	0.432653	32.876451	0.0
1	(madame)	(defarge)	0.013160	0.019307	0.008353	0.634731	32.876451	0.0
2	(manette)	(doctor)	0.010954	0.015366	0.005280	0.482014	31.368013	0.0
3	(doctor)	(manette)	0.015366	0.010954	0.005280	0.343590	31.368013	0.0

907 `new_book3_rules_min_005`

907

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	lev
6	(organic)	(beings)	0.009536	0.008058	0.005544	0.581395	72.151696	0.0
7	(beings)	(organic)	0.008058	0.009536	0.005544	0.688073	72.151696	0.0
1	(natural)	(selection)	0.023287	0.024691	0.016190	0.695238	28.157143	0.0
0	(selection)	(natural)	0.024691	0.023287	0.016190	0.655689	28.157143	0.0
3	(life)	(conditions)	0.019221	0.014859	0.006432	0.334615	22.519116	0.0
4	(conditions)	(life)	0.014859	0.019221	0.006432	0.432836	22.519116	0.0
8	(genus)	(species)	0.008871	0.094773	0.005397	0.608333	6.418818	0.0
9	(genera)	(species)	0.013159	0.094773	0.005249	0.398876	4.208737	0.0
5	(distinct)	(species)	0.015303	0.094773	0.005766	0.376812	3.975921	0.0

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	lev
2	(varieties)	(species)	0.027722	0.094773	0.007467	0.269333	2.841866	0.0

Now we finally have some strong and interesting words that we can look at. And the processes of which that we can go through to find such strong interesting rules. First lets talk about these rules.

Book1: There is clearly a "lady middleton" and "Colonel Brandon". The strong lift values indicate that these two words appear together a lot. Looking at the conviction shows us that the probability of middleton appearing without lady, and brandon appearing without colonel is higher. Although I have not read the book, this makes sense. The author probably does not always say lady middleton or Colonel brandon everytime, but rather middleton or brandon. Either way these association rules help us indicate that there is a lady middleton and a colonel. Lastly, it looks like elinor said a lot of things throughout the book. Or at least the word elinor and said appear together with a lift value of 3.744. This is interesting.

Book2: madame degrage, and doctor manette are two main characters. They have high lift values showing us that the two characters a madame and a doctor.

Book3: "organic beings", "natural selection," and "life conditions" are clearly things that Darwin talks about a lot within his book. This super interesting, they have really high lift values and are strong rules with large confidence rules. The same goes for talking about genus species, distinct species and the varieties of species. All things that we would expect to be in this book.

Final Remarks

This has been a fun exercise in looking using the apriori algorithm to look at the contents of three different books. Throughout the processes we looked at the books with stop words that were included. If we were to compare these books to a book in 2020 we might be able to see the difference in stop word usage. However, the three books were written in the same century.

To make the data look a bit more meaningful we had to remove the stop words and ['mrs','mr','s','t','sir','miss']. When this occurred we lost a lot of the association rules. The reason being is that we were looking at text books. Not every one of the words is going to appear in every sentence. By lowering the min_support, we were able to find more words that appeared together. In book1 & 2 this left use with more interesting characters such as colonel brandon and madame defarge. I'm positive if we left in the titles we would have seen more characters, but on the otherhand I was stripping out the title's to try and see if I could find more meaningful rules such as "elinor said". Book3 did not offer as many of these problems because darwin was talking

about different species and organic beings. But we could continue the mining processes to see if we could find different meaningful associations rather than words that are used together to describe something. T and S were taken out because the data alread had "'" removed. And just seeing possessions associated with other words was as much interest to me.

The main things that were looked at was the lift. With a lift greater than 1.1 we were determining if the rules were interesting. The Support values showed us how often these words appeared. and the confidence showed how confident we are that the word appeared together. Conviction shows us the probability of x occuring without y.