INF2209 Project Write Up (Updated)

Introduction – Description of Dataset

The Amazon Product Review dataset I used in this assignment is extracted from Kaggle (Data Source Link: Amazon Product Review Dataset Link). The dataset has 1597 records and it originally contains 27 columns, which are ID, product id asins, product brand, product category, product color, product update date, dimension of the product, EAN Code of the project, keys, product manufacturer, manufacturer number, product name, price, review date, product recommendation, number of helpful upvotes for reviews, product ratings, review sources, product review, review title, product review user city, review user province, product review user username, product size, product UPC, and product weight. More specifically, for the purpose of performing topic modelling, we will only focus on the textual columns.

Methods, Results and Conclusion

1. Data Cleaning

(1) Select only Text Columns

Since the goal of this analysis is to perform the topic modelling, we will focus only on the textual columns in the database, and then drop all the other numeric or categorical columns. The Selected columns for Amazon product review dataset are “name”, “reviews.username”, “reviews.rating”, “reviews.title”, and “reviews.text”.

(2) Make text lowercase

(3) Remove text in square brackets

(4) Remove punctuations

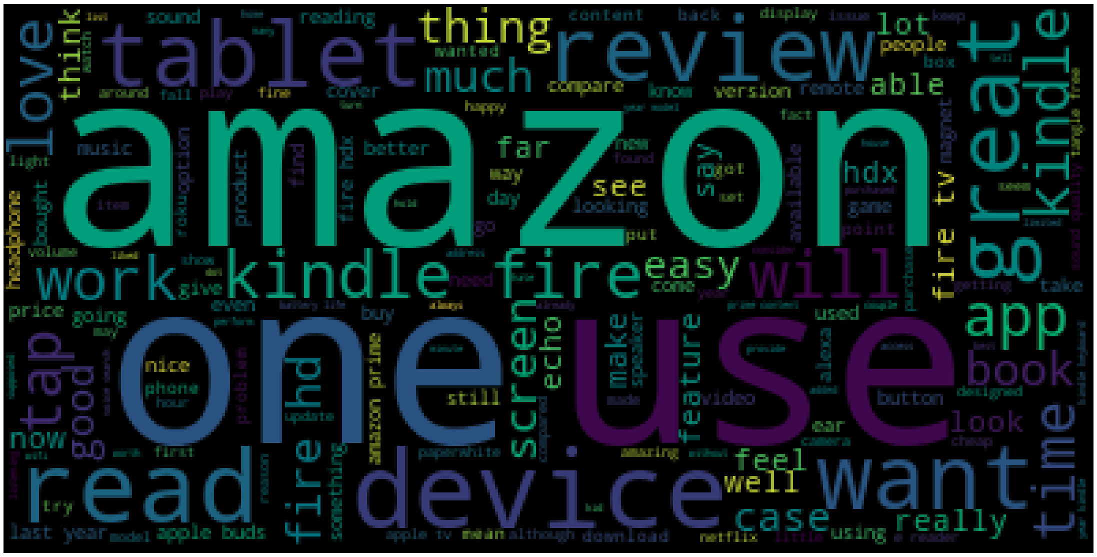
(5) Remove read errors

(6) Remove words containing numbers or special characters

2. Exploratory Analysis

(1) Word Cloud

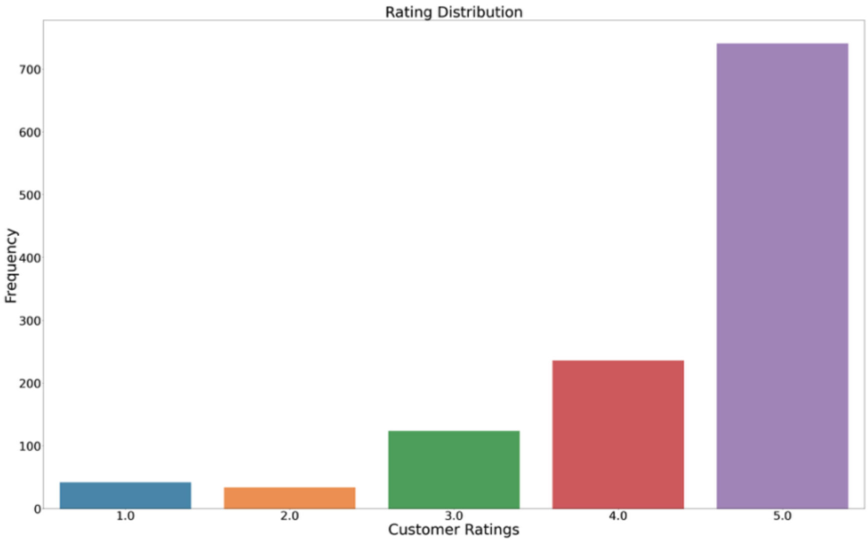
We will create a word cloud to visually represent the most common words from the text columns to understand the data and ensure whether the preprocessing is ready.



From the word cloud we have generated, we could observe the top 100 common words from the review text column in the dataset. Through ranking the words by size, we could see that the most 10 common words in the reviews text are “amazon”, “one”, “use”, “read”, “device”, “tablet”, “review”, “great”, “kindle”, and “fire”.

(2) Bar Plot for Customer Rating Distribution

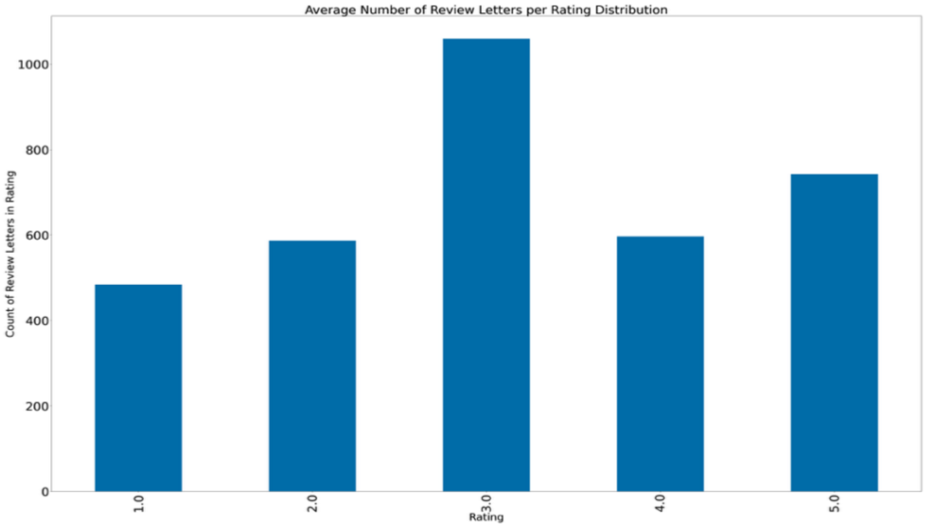
We are creating the histogram plot for ranking the counts for each rating hierarchy.



From the histogram, we can see that around 700 users are making the highest rating, 250 users are rating 4 stars, and about 200 users are rating 3 stars or below for the products.

(3) Bar Plot for Average Number of review letters per Rating Distribution

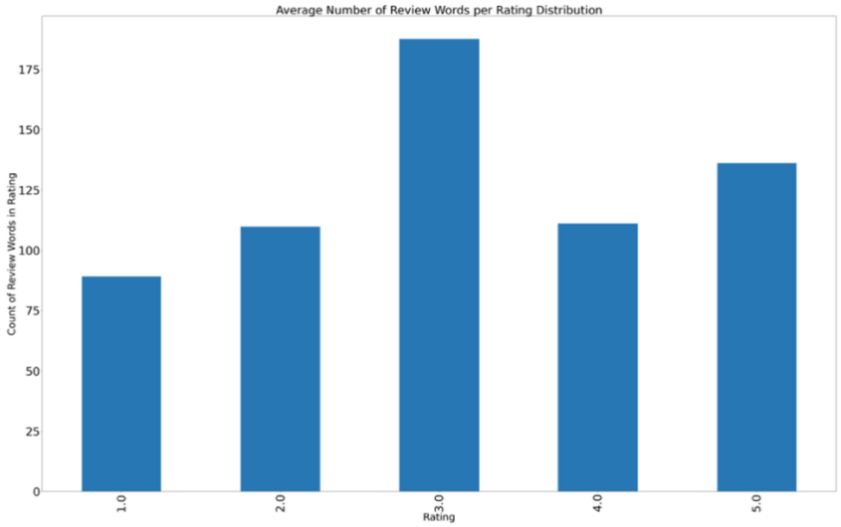
We created the new column named as ‘review_len’ to indicate the length of letters for each review, and then we will calculate the average number of review letters per each rating hierarchy and generate the histogram.



From the plot, we could see that users which rating with 3 are having the most letters of reviews on average. Rating with 2 and 4 are having the similar number of letters of review. Users with the lowest rates are also having the least number of review letters.

(4) Bar Plots for Average Number of review words per Rating Distribution

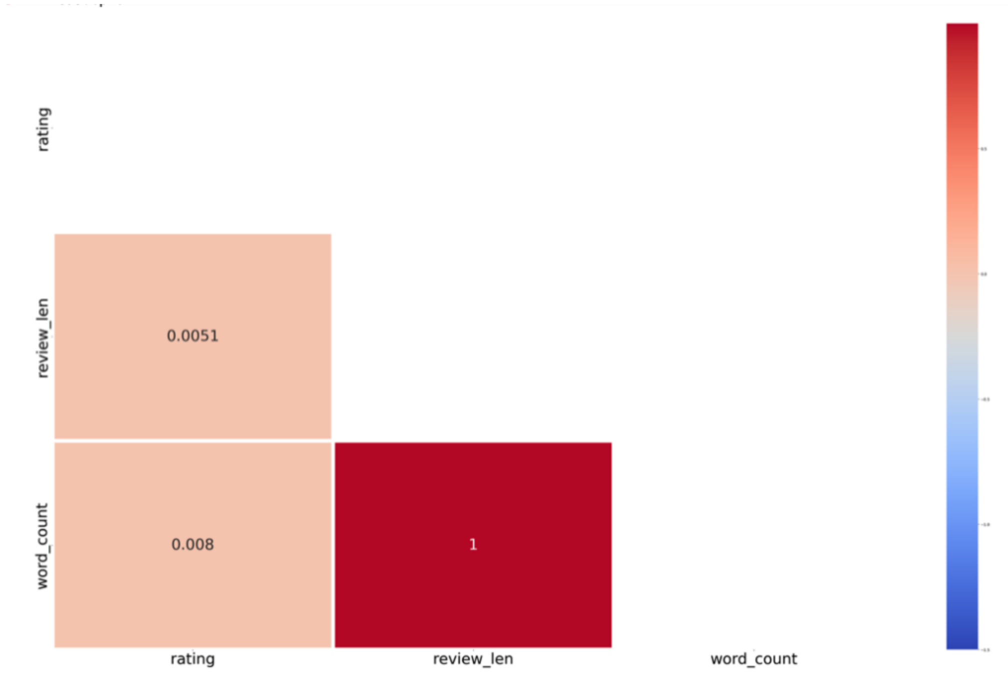
Similarly, we also created the new column named as ‘word_count’ to indicate the length of words for each review text, and then we will calculate the average number of review words per each rating hierarchy and generate the histogram.



The histogram plot representation for word counts per rating level shows the same as the letter counts'. Users rating with 3 has around 175-words reviews on average and users only write less than 100 words averagely while they rate with 1 for the products.

(5) Correlation Plot Between Rating and Word Counts of Reviews Text

We will also take a look at the correlation between rating level and their review length regarding the letter counts and word counts of review texts.



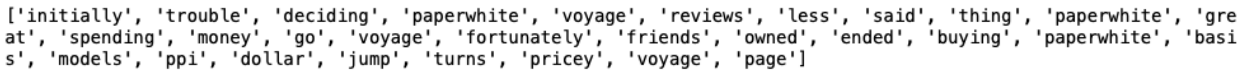
From the correlation plot, the correlation between rating and word counts of reviews is 0.008 and the correlation between rating and letter counts of reviews is 0.0051. The values of correlation are extremely low, which means that there is not really relationship between rating levels and word counts of reviews.

3. Topic Modelling Method

Method 1: Latent Dirichlet Allocation (LDA)

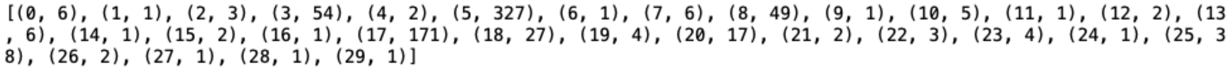
(1) Data preparation

i. Transform the textual data in a format as an input for training LDA models



```
['initially', 'trouble', 'deciding', 'paperwhite', 'voyage', 'reviews', 'less', 'said', 'thing', 'paperwhite', 'gre
at', 'spending', 'money', 'go', 'voyage', 'fortunately', 'friends', 'owned', 'ended', 'buying', 'paperwhite', 'basi
s', 'models', 'ppi', 'dollar', 'jump', 'turns', 'pricey', 'voyage', 'page']
```

ii. Term Document Frequency



```
[(0, 6), (1, 1), (2, 3), (3, 54), (4, 2), (5, 327), (6, 1), (7, 6), (8, 49), (9, 1), (10, 5), (11, 1), (12, 2), (13
, 6), (14, 1), (15, 2), (16, 1), (17, 171), (18, 27), (19, 4), (20, 17), (21, 2), (22, 3), (23, 4), (24, 1), (25, 3
8), (26, 2), (27, 1), (28, 1), (29, 1)]
```

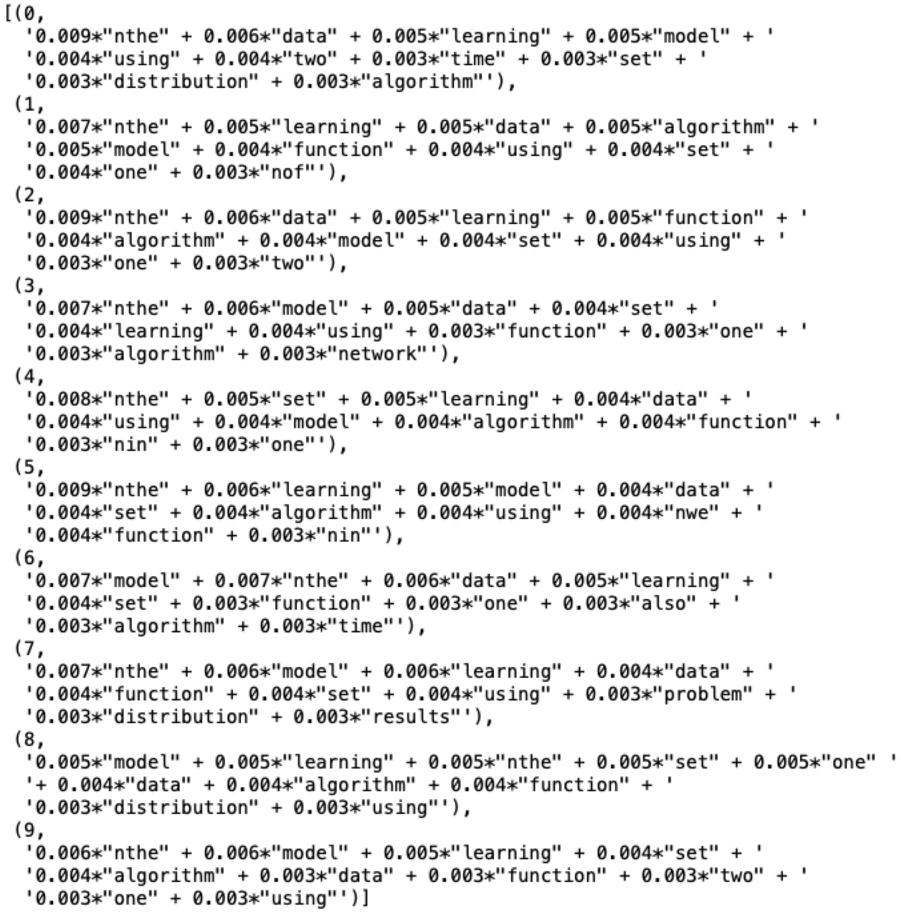
(2) LDA Model Training

For the comparison with NMF models, I will also build the LDA model with 8 topics where each topic is a combination of keywords, and each keyword will contribute a specific weightage to the topic.

Training Result:

The perplexity of the LDA model is -7.0946 which indicates that the model describes a set of documents well.

The keywords in the 10 topics are shown as following.



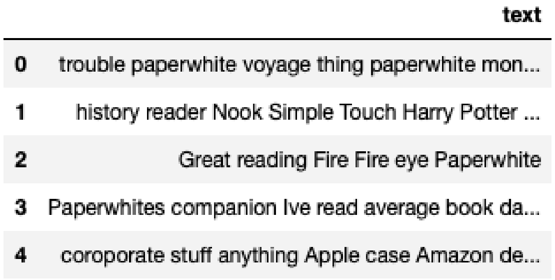
```
[(0,
  '0.009*"nthe" + 0.006*"data" + 0.005*"learning" + 0.005*"model" + '
  '0.004*"using" + 0.004*"two" + 0.003*"time" + 0.003*"set" + '
  '0.003*"distribution" + 0.003*"algorithm"'),
 (1,
  '0.007*"nthe" + 0.005*"learning" + 0.005*"data" + 0.005*"algorithm" + '
  '0.005*"model" + 0.004*"function" + 0.004*"using" + 0.004*"set" + '
  '0.004*"one" + 0.003*"nof"'),
 (2,
  '0.009*"nthe" + 0.006*"data" + 0.005*"learning" + 0.005*"function" + '
  '0.004*"algorithm" + 0.004*"model" + 0.004*"set" + 0.004*"using" + '
  '0.003*"one" + 0.003*"two"'),
 (3,
  '0.007*"nthe" + 0.006*"model" + 0.005*"data" + 0.004*"set" + '
  '0.004*"learning" + 0.004*"using" + 0.003*"function" + 0.003*"one" + '
  '0.003*"algorithm" + 0.003*"network"'),
 (4,
  '0.008*"nthe" + 0.005*"set" + 0.005*"learning" + 0.004*"data" + '
  '0.004*"using" + 0.004*"model" + 0.004*"algorithm" + 0.004*"function" + '
  '0.003*"nin" + 0.003*"one"'),
 (5,
  '0.009*"nthe" + 0.006*"learning" + 0.005*"model" + 0.004*"data" + '
  '0.004*"set" + 0.004*"algorithm" + 0.004*"using" + 0.004*"nwe" + '
  '0.004*"function" + 0.003*"nin"'),
 (6,
  '0.007*"model" + 0.007*"nthe" + 0.006*"data" + 0.005*"learning" + '
  '0.004*"set" + 0.003*"function" + 0.003*"one" + 0.003*"also" + '
  '0.003*"algorithm" + 0.003*"time"'),
 (7,
  '0.007*"nthe" + 0.006*"model" + 0.006*"learning" + 0.004*"data" + '
  '0.004*"function" + 0.004*"set" + 0.004*"using" + 0.003*"problem" + '
  '0.003*"distribution" + 0.003*"results"'),
 (8,
  '0.005*"model" + 0.005*"learning" + 0.005*"nthe" + 0.005*"set" + 0.005*"one" '
  '+ 0.004*"data" + 0.004*"algorithm" + 0.004*"function" + '
  '0.003*"distribution" + 0.003*"using"'),
 (9,
  '0.006*"nthe" + 0.006*"model" + 0.005*"learning" + 0.004*"set" + '
  '0.004*"algorithm" + 0.003*"data" + 0.003*"function" + 0.003*"two" + '
  '0.003*"one" + 0.003*"using"')]
```

Method 2: Non-Negative Matrix Factorization (NMF)

(1) Data preparation:

i. Noun extraction and lemmatize function

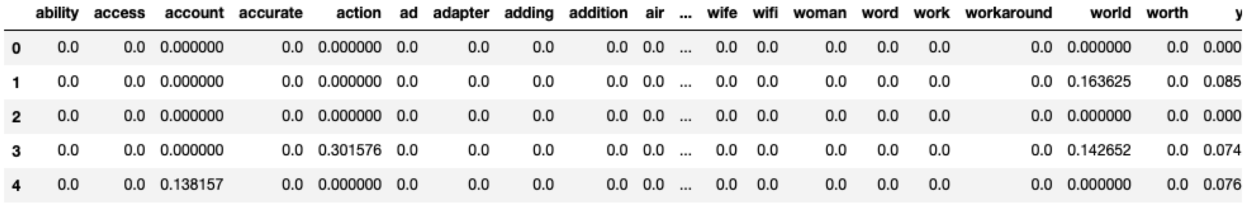
We will pull out only the nouns from the review text and then tokenize(lemmatize) the text. The figure below shows the sample rows after filtering the text with only nouns.



	text
0	trouble paperwhite voyage thing paperwhite mon...
1	history reader Nook Simple Touch Harry Potter ...
2	Great reading Fire Fire eye Paperwhite
3	Paperwhites companion Ive read average book da...
4	coroporate stuff anything Apple case Amazon de...

ii. Term-document Matrix

We will fit and transform review noun text to a TF-IDF Document-Term matrix.



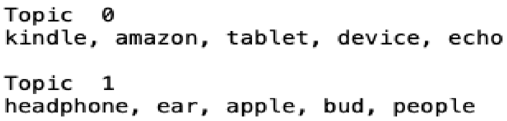
	ability	access	account	accurate	action	ad	adapter	adding	addition	air	...	wife	wifi	woman	word	work	workaround	world	worth	y
0	0.0	0.0	0.000000	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.000
1	0.0	0.0	0.000000	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.163625	0.0	0.085
2	0.0	0.0	0.000000	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.000
3	0.0	0.0	0.000000	0.0	0.301576	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.142652	0.0	0.074
4	0.0	0.0	0.138157	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.076

(2) NMF Model Training

For ease of comparison and understanding, we will look at 2 topics and 8 topics that model has generated.

Training Results:

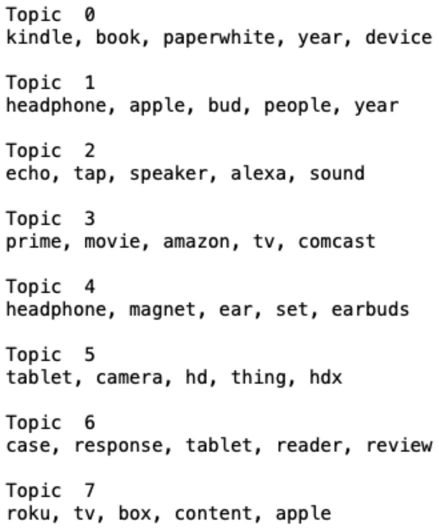
With 2 topics:



```
Topic  0
kindle, amazon, tablet, device, echo

Topic  1
headphone, ear, apple, bud, people
```

With 8 topics:



```
Topic  0
kindle, book, paperwhite, year, device

Topic  1
headphone, apple, bud, people, year

Topic  2
echo, tap, speaker, alexa, sound

Topic  3
prime, movie, amazon, tv, comcast

Topic  4
headphone, magnet, ear, set, earbuds

Topic  5
tablet, camera, hd, thing, hdx

Topic  6
case, response, tablet, reader, review

Topic  7
roku, tv, box, content, apple
```

Discussion & Analysis

From the results using LDA and NMF model trainings, we can clearly see that NMF produced more relative topics compared to LDA. Most of the entries are close to zero and only a few parameters have significant values in NMF models. The keywords generated from LDA and NMF are ‘headphone’, ‘tablet’, ‘amazon’, ‘device’, ‘function’, ‘great’ and so on, which indicate that the most product categories people wrote the reviews with are technical devices and most people are making positive reviews. Therefore, for the amazon product review dataset, we can clearly see that most people are making positive reviews and people love buying technical device through Amazon.