Amazon Office Product Reviews' Analysis

Ayesha Aziz, Ryan Siegel, Sushant Jha, CareyAnne Howlett

GITHUB LINK TO CODE:
https://github.com/rwsiegel/DS5230_UML_Project

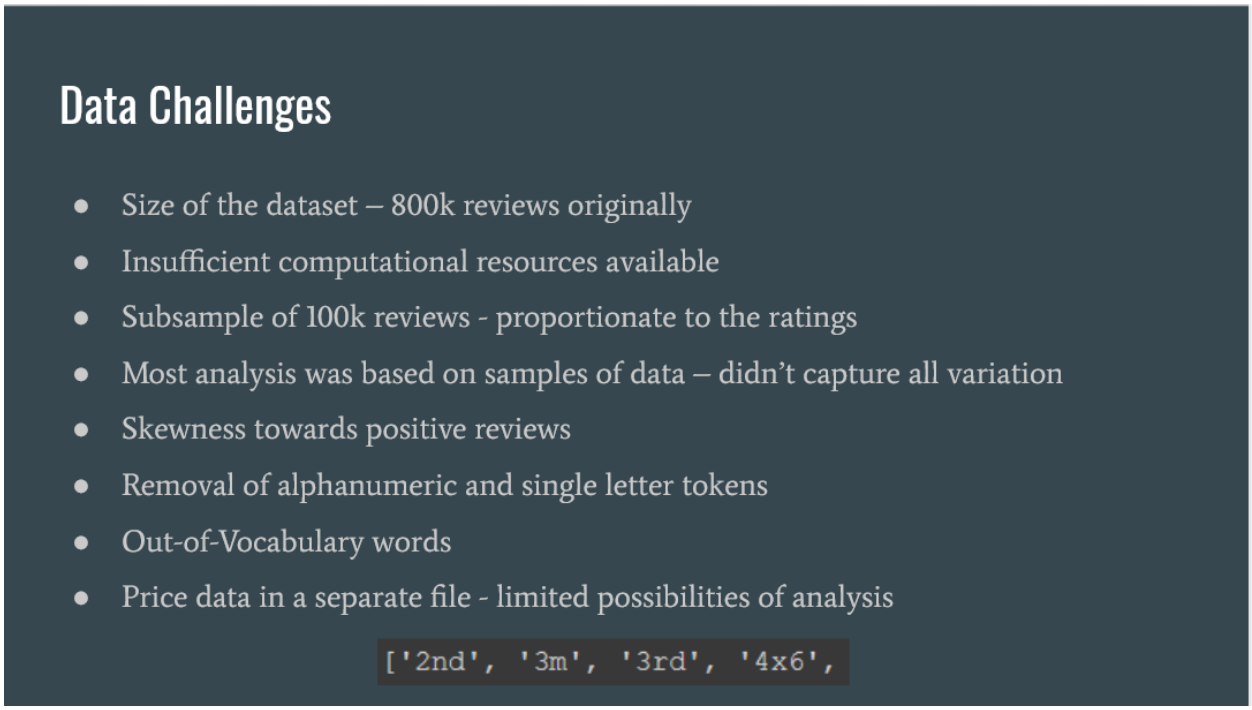
Outline

- Introduction
- Data Challenges
- Exploratory Data Analysis
- Clustering
- Text Processing
- Topic Modeling - Latent Dirichlet Allocation and Latent Semantic Analysis
- Additional Analysis
- Replication in Other Categories

The dataset that was used for this project is called the Amazon Office Product Reviews dataset. This dataset is actually a subset of the Amazon Product Reviews dataset that comprises over 82 million reviews. With the interest of time and resources, this project focused on the Office Product portion of the Amazon Product Reviews dataset. The Amazon Office Product Reviews dataset contains over 800,000 reviews of office products that are available for purchase on Amazon. The reviews that make up this dataset are 5-core reviews meaning that the products in the dataset have at least 5 reviews that were written by users who have written at least 5 reviews using the same Amazon account.

With the Amazon Office Product Reviews dataset, the objective is to determine what characteristics are most important to people buying these products. In turn, this information can be used to filter through different products that someone is looking for and recommend those that highlight these important characteristics.

To find these important characteristics, the reviews will go through text processing, and then will be analyzed by topic modeling algorithms. Some of the topic modeling algorithms that will be used are Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA). Clustering algorithms such as Principal Component Analysis (PCA) and t-distributed Stochastic Neighbor Embedding (t-SNE) will also be used to find groups of similar reviews.



Though the Amazon Office Product Reviews dataset is publicly available, there were still some data challenges. For example, the size of the dataset was too large for the algorithms we ran. This resulted in further subsetting the dataset to 100,000 reviews. Because the algorithms were only able to run on 100,000 review subsets, the results were not able to capture the whole scope of the variation in the dataset.

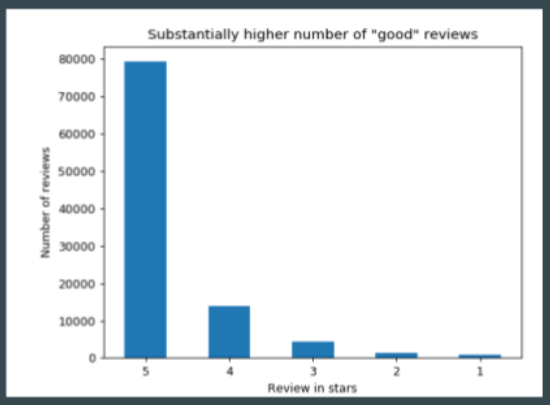
When conducting the exploratory data analysis, it was found that the data is skewed toward positive reviews than negative. With skewed data, it makes it more challenging to draw conclusions and generate groups based on “good” and “bad” reviews. With mostly positive reviews, it’s hard to determine which characteristics of the products are most important to the reviews.

Another problem that came up was with text preprocessing. In order to get the best results, heavy text preprocessing was necessary. This included removing stop words, expanding contracted words, and removing punctuation. However, problems arose while trying to remove alphanumeric and single letter “words” (i. e. “n” and “t”). In order to remove the alphanumeric tokens, additional steps were taken to remove all words that were not in the English dictionary. However some of the single letter/few out-of-vocabulary tokens were unsuccessfully taken out of the text data. In future work, more time could be used to fix these issues.

There was additional metadata with this dataset that contained information about the price of these products. It also included other products reviewers were buying in addition to the products they left a review for. However, the metadata was too large to access therefore it was not included in the project analysis.

Exploratory Data Analysis

- Total of over 800k rows categorized into 12 attributes
- Over 95% of all reviews are positive
- ~ 78% of all the reviews fall under the verified category
- Just over 12% of all reviews posted by unique users
- Negligible proportion of missing values under text reviews



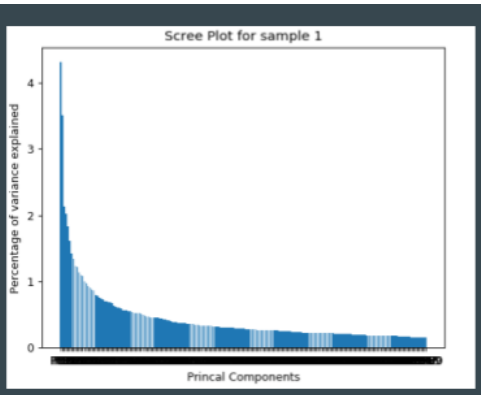
As mentioned before, the final data set selected for the work is the Amazon Office Product Reviews in which each product has been reviewed by at least 5 customers and each customer has posted a minimum of 5 reviews on Amazon. This filtering criteria was essential to uphold the integrity of the reviews posted and used for analysis. Divided into a total of 12 columns, although there were considerable amounts of missing values in some attributes of the data set, this analysis primarily focuses on raw text reviews. The column for test reviews has a negligible proportion of missing values, and the rows corresponding to these observations were eliminated from further analysis.

The initial visualizations using the complete data set revealed that the reviews were highly skewed towards the positive side as reflected in the bar chart. A major proportion of the reviews posted were positive in nature (in terms of the star ratings), consequently leading to the imbalance in data with respect to this attribute. This might indicate a potential survivorship bias as naturally, better products would stay on Amazon more than poorly reviewed and presumably low quality products. During further analysis, sampling of the data was performed using the same proportions in check.

Another observation that came to light from the exploratory data analysis phase was that out of all the reviews posted, only 12.68% were posted by unique reviewers. This does not come as a surprise, as by intuition, we estimate only a small percentage of all the shoppers to actually post reviews. Also, it is more likely that an active reviewer posts more reviews. All other attributes associated with each review were examined to familiarize us with the data in more depth before moving on to the next phase of the work.

Dimensionality Reduction

- PCA and T-SNE employed for dimensionality reduction
- T-SNE did not return useful results
- Scree plot was generated using PCA
- Over 400 principal components explained for only 62% variance in the data
- Using a sum-sample of 100K rows, it could explain for over 80% of the variance



Considering the enormously large number of dimensions in all text reviews, it was only obvious to try and perform dimensionality reduction on them to see if we could get a relatively lower number of dimensions (e.g. principal components) that could explain for most of the variance in the data. The techniques of Principal Component Analysis (PCA) and T-SNE were performed on the dataset.

A sub-sample of 100,000 observations was used to perform T-SNE since running this algorithm on the complete data set is computationally very expensive and we were limited by the resources. The result outputted by T-SNE was not very useful and/or interpretable. One of the potential reasons for the algorithm's failure here is that it is limited by the number of components. In practice, we could only specify a maximum of 4 components as a parameter to the T-SNE function used.

Principal component analysis, on the contrary, performed better here. From the scree plot generated using PCA on the complete data set (800,000 rows), it was observed that only around 62% of the variance could be explained by over 400 components. When performing on a random subset of 100,000 samples, this proportion of explained variance rose to ~80%. This illustrates one of the downfalls of sampling, in that we would have a data set with less variation in general.

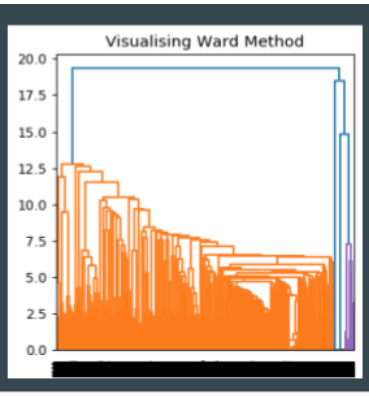
Clustering

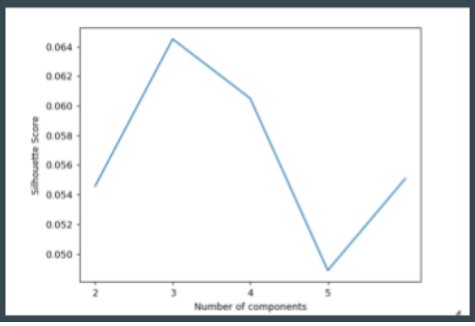
Following clustering methods were performed on the processed review text to determine if there were any strong clusters:

- K-means
- Gaussian Mixture Model
- Dbscan

The results were not fruitful, silhouette scores were mediocre. K-means performed the best, still only at ~6% on average.

- Lended further credence to Topic Modeling.





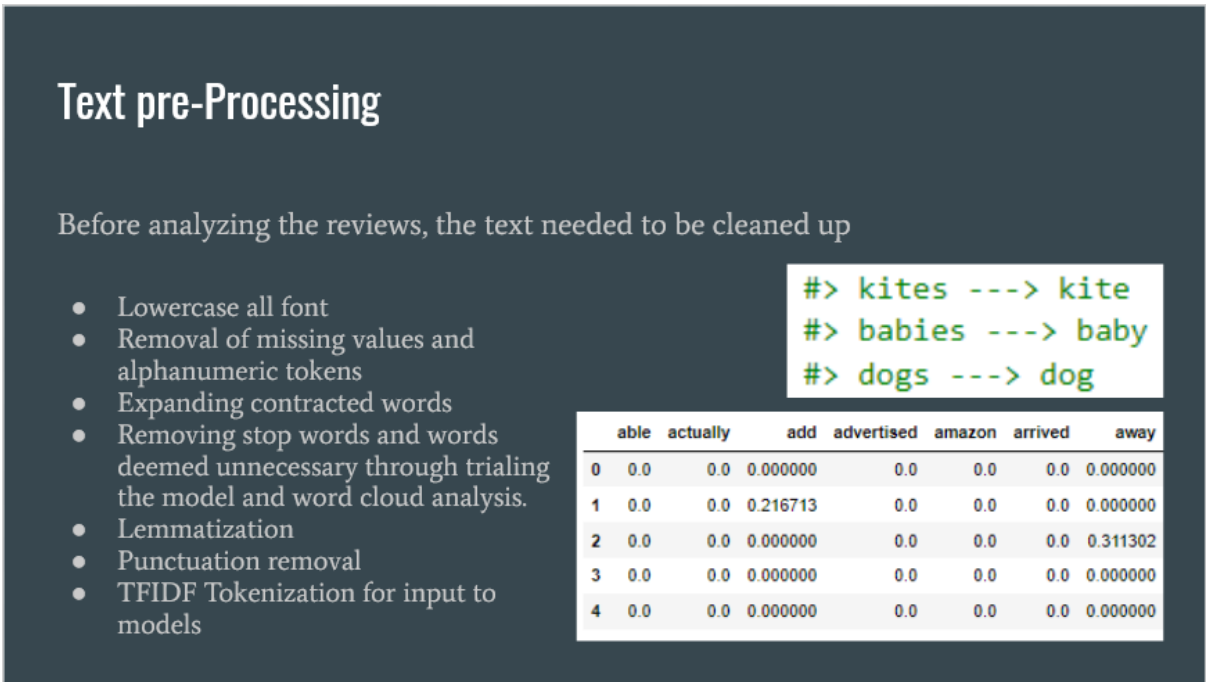
While trying to understand the data as a whole, the idea was to try and see if any clusters could be found on existing review data in its raw form. It could be that due to the imbalance of reviews towards the positive ratings that clustering that could arise from a positive / negative standpoint was hidden due to the imbalance data set.

For the purpose of clustering, three different types of clustering techniques were employed:

1. Centroid based clustering (K-means)
2. Distribution based clustering (GMM)
3. Density based clustering (DBSCAN)

The results generated by all of the aforementioned techniques were not encouraging in the sense that none of the algorithms could output consistent and meaningful clusters. DBSCAN performed the worst among all the employed methods. K-Means, a centroid centered algorithm, performed better than the other two. However, even K-Means could generate clusters with an average Silhouette score of about 6%, which is substantially low.

The failure of these clustering techniques paved the way for the deployment of Topic Modeling on the data set being used. Most telling, when viewing the dendrogram for GMM, applied on 10000 reviews, no meaningful clusters were generated. Objectively speaking as well, when looking at CH Index and Silhouette Scores generated, it cemented the need to do topic modeling to best learn about the review data.



When doing text pre-processing, what we are looking to do is to remove words that would otherwise create more dimensions within the text documents and little information. In order to do this there are several different techniques we must apply to the data and within a specific order. It is virtually doing dimension reduction and bucketing specific words when they are lemmatized.

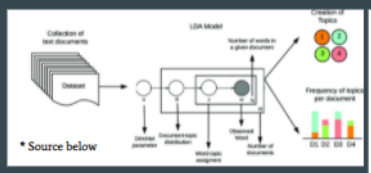
Like other parts of the presentation, we lower the font of the text since string text is interpreted differently when it is capital case or lowercase. Afterwards, numbers are removed as it is difficult to interpret numbers in the context of a review. In addition, there are characters such as ‘1aa’ or odd pairings of letters/words with numbers and punctuation to clean. The next step is to expand contractions out such as don’t or haven’t to ‘do not’ or ‘have not’. Without doing this, it is difficult to do lemmatization as well as removing stop words.

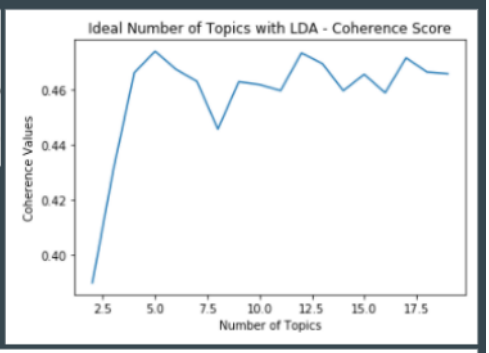
Next we remove stop words, which is from a set dictionary of words like ‘as, if, the’ since they are common and do not provide more information on the document. Included in this step is also the removal of words we have deemed overwhelming to the topic modeling, a process learned **after** the model is run. We are taking prior knowledge and making specific adjustments. More on this within the word cloud CareyAnne will present. An important note, we did not use stemming as this was hurting our model quality. Through research, we found other suggestions online that stemming is not always suited for the task at hand. We used discretion and skipped it here.

Stop words and miscellaneous text is removed again as an additional data cleaning step since we kept finding some words that ‘should’ have been cleaned but were not. The last step is removing punctuation to refine the data set then tfidf tokenization for the models and analysis.

Topic Modeling - Latent Dirichlet Allocation (LDA)

- By taking the pre-processed data, we would then utilize LDA to surface key topics (or a vector set of words) of the collection of documents.





- We would help understand the best model through a combination of studying the output and through Coherence Scoring. We would then take these topics and via the LDA model, assign probabilities to each review being associated with a particular topic.

- With these probability assignments, we could then group on the product itself and find the mean probabilities of all topics with a particular amazon Product.

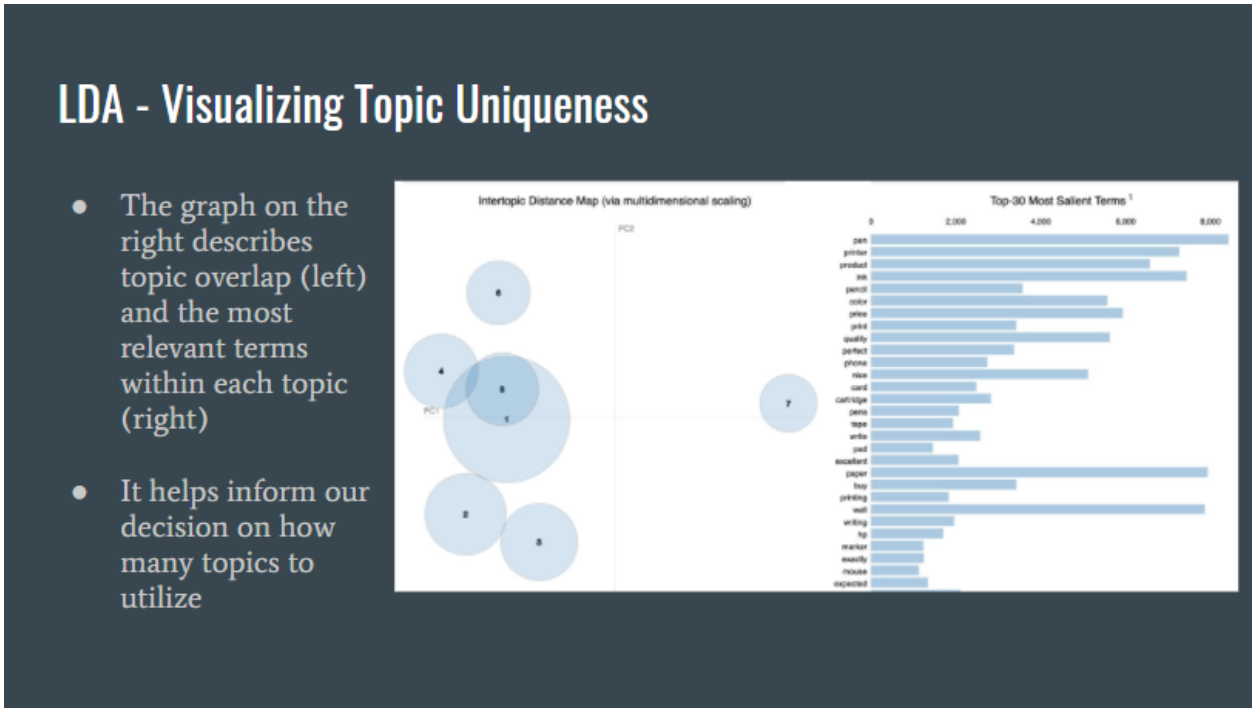
```
(0, '0.065*"paper" + 0.044*"quality" + 0.043*"price" + 0.037*"well" + 0.026*"perfect"')
(1, '0.106*"pen" + 0.044*"ink" + 0.044*"color" + 0.030*"write" + 0.026*"pens"')
(2, '0.036*"one" + 0.030*"use" + 0.029*"pencil" + 0.023*"phone" + 0.020*"card"')
(3, '0.026*"it" + 0.022*"one" + 0.019*"use" + 0.015*"get" + 0.014*"little"')
(4, '0.107*"product" + 0.053*"i" + 0.032*"a" + 0.029*"m" + 0.027*"price"')
(5, '0.079*"printer" + 0.042*"ink" + 0.037*"print" + 0.031*"cartridge" + 0.020*"printing"')
```

* https://www.researchgate.net/profile/Diego-Buenano-Fernandez/publication/339368709/figure/fig1/AS:860489982689280@1582168207260/Schematic-of-LDA-algorithm.png

What we set out to do overall was to answer the question “Does the product I am looking for match the characteristics I think are important?”.

The idea here is to take the pre-processed text data and utilize LDA to summarize pieces of information per product (in this case through topic probability distributions). To execute this, we implemented an LDA model that had the highest coherence score we could achieve. Coherence scoring is a more objective measure than ‘eye-balling’ the topics to determine what would actually be the ideal number for a given document matrix. The method we used is ‘c_v’ which fundamentally is a particular way to understand the log likelihood of a pair of words showing up together within a particular document given the corpus (conceptually known as normalized pointwise mutual information).

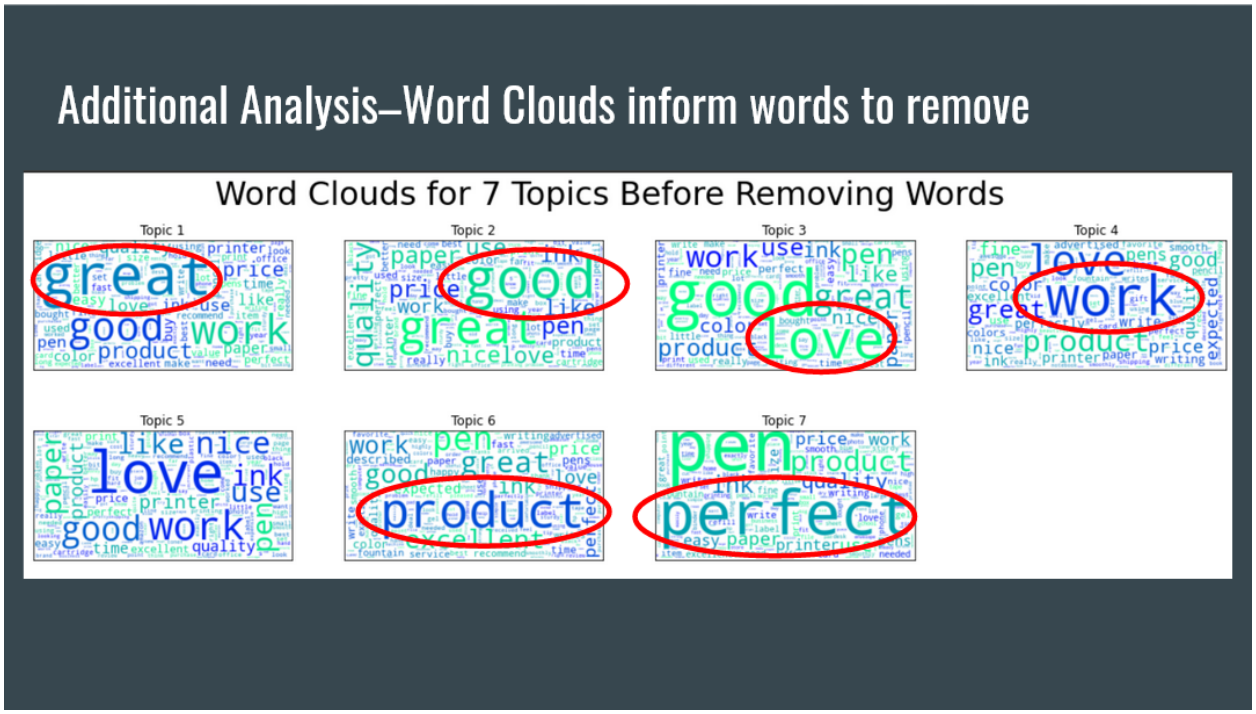
With a set of coherence scores outlined for various topic counts, we pick a higher scoring topic count that would still provide a diverse set of words within the topic. Here we can see the topics forming, such as topic 0 described as a high quality and well priced product, likely having to do with paper. These topics are then assigned with associated probabilities at the review level.



This graph on the right was generated from the pyLDAvis python package. It helps describe the overall topic uniqueness on the right by using Principal Component 1 and Principal Component 2. It also gives the top 30 most relevant words predicted within each topic. This gives a guide to which words are relevant to each topic in terms of importance.

With a set of coherence scores outlined for various topic counts, we pick a higher scoring topic count that would still provide a diverse set of words within the topic. Here we can see the topics forming, such as topic 0 described as a high quality and well priced product, likely having to do with paper. These topics are then assigned with associated probabilities at the review level.

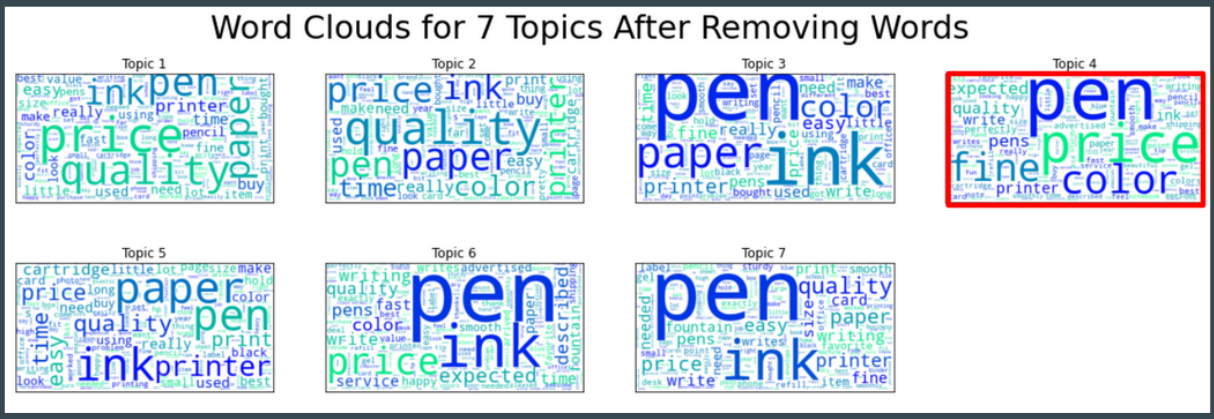
These assigned probabilities are used to then augment our original dataset for further analysis. We then group by each of the products by their reviews and find the mean probability for each set of product reviews as it relates to all topics.



Word clouds were a great way to visually show the importance of each word in the 100,000 review subset to the 7 topics determined through Latent Semantic Analysis (LSA). The words that were included in the word clouds were ones that appear 1,000 or more out of all of the reviews (in the subset). This meant that out of 31,194 unique words used in the reviews, only 272 appeared 1,000 times or more. From there, the LSA algorithm generated weights for each word relative to each topic determined by the algorithm (a hyperparameter was set so that 7 topics were generated).

From there, 7 word clouds were created for each topic. The weight of the relation of the word to the respective topics was used as the mechanism to determine word importance. So in the images, the bigger the word is, the larger the relation it has to the given topic. In this array of word clouds, the “bigger” words consisted of words that didn’t help describe the topic or add any value to the analysis. Words like “great”, “good”, and “love” aren't telling of what qualities are most important to people who are reviewing these products.

Additional Analysis–Word Clouds after removing non-useful words

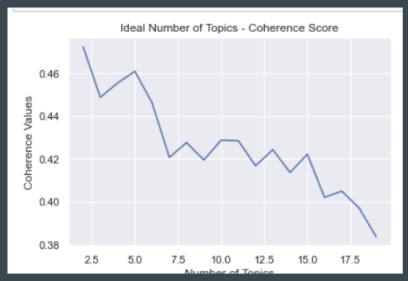


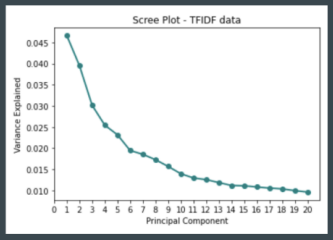
After determining which words were not descriptive of these topics, they were removed from the reviews all together. Once they were removed, the word clouds were replotted resulting in the array of word clouds here. After removing the non descriptive words, it is easier to see which words or characteristics were most important to the reviewers.

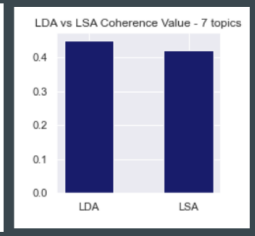
For example, while looking at Topic 4, the highly correlated words are “pen”, “price”, and “color”. This suggests that this topic is about pen and the qualities that are most important to the reviews! This suggests that while people are looking for pens, they are looking at the price and color of them.

Topic Modeling - LSA

- Performed using gensim and separately with SVD
- LDA performed marginally better
- Interesting similarities between terms and topics







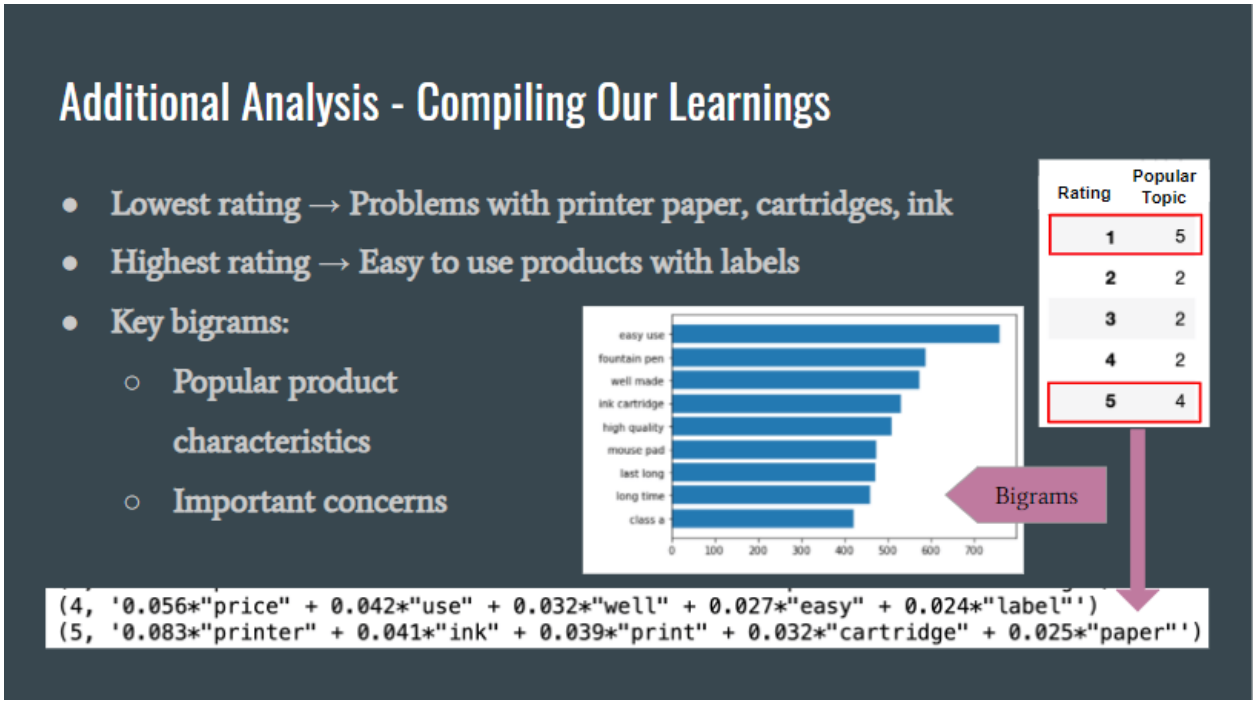

```
(0, '0.369*"printer" + 0.273*"one" + 0.267*"ink" + 0.241*"paper" + 0.237*"use" + 0.193*"pen" + 0.191*"print"')
(1, '0.616*"pen" + -0.597*"printer" + -0.258*"print" + -0.137*"printing" + -0.108*"cartridge" + 0.099*"writing" + 0.091*"use"')
```

Next, Topic Modeling was performed via Latent Semantic Analysis, which utilizes dimensionality reduction on a document-term matrix, and presents terms as linear combinations of each other. The result isn't the same as LDA's (which directly creates topics out of a doc-term matrix), as LSA revolves around finding the cosine similarities between terms and documents, resulting in a number from -1 to 1, indicating how strongly correlated the two are.

Earlier on, while performing clustering, we noticed how dimensionality reduction did not bear fruit. The same can be seen here in the scree plot, which shows that 20 dimensions do not explain even 50% of the variation in the dataset. However, owing to the lack of means to measure Truncated SVD's performance (since we are dealing with text data, and we lose textual information in dimensionality reduction), we resorted to using LSA from gensim - a model that shows the top terms against each document, and the correlation instead of probability.

From gensim, the coherence scores over 20 topics indicated towards 2 being the ideal number of clusters/topics to keep. The top 7 terms from the top 2 topics generated hint towards the first one being about 'printers, inks and papers', and the second one being about 'pen usage in writing'. While they don't tell much, it's interesting to see how different themes can emerge when using cosine similarity.

To compare performance with LDA, however, we ran it for 7 topics and noticed that LDA performed slightly better, which is why most of our analysis was based on LDA. It's interesting to note, however, that LSA uncovered a theme surrounding 'writing', which we didn't see for LDA.

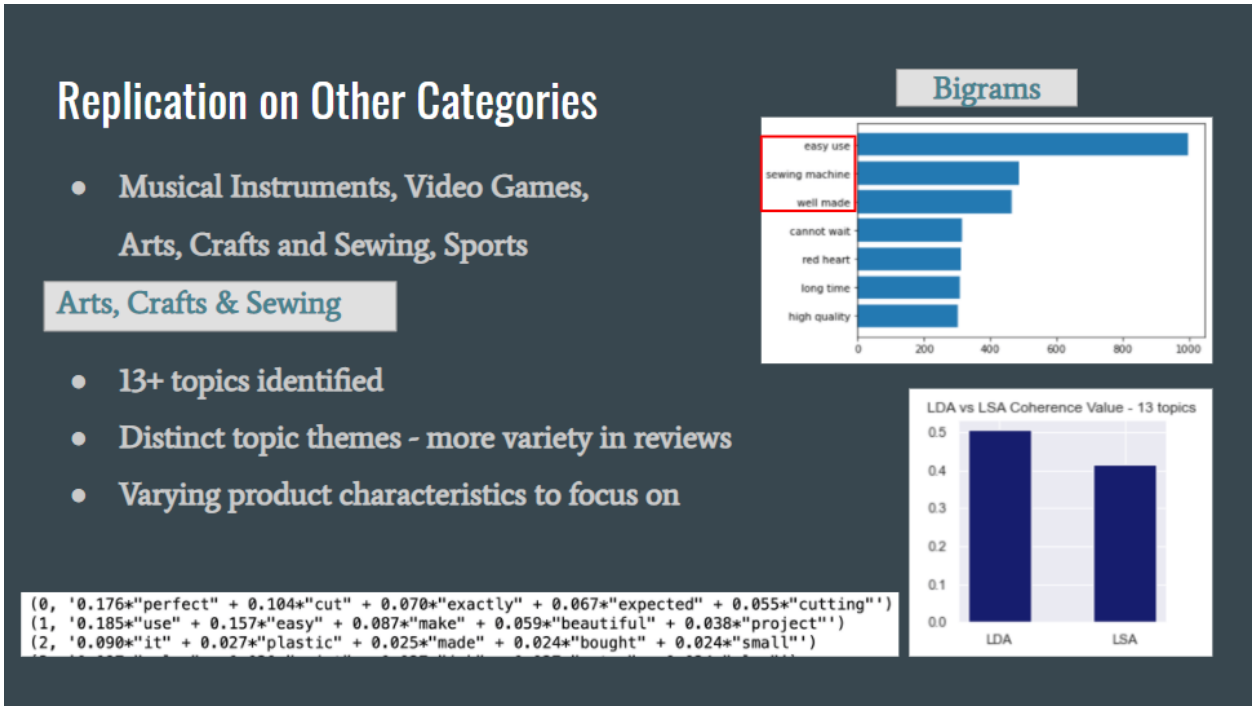


After cleaning the data and removing overused words that are surfaced within the word clouds, we were able to then focus on more takeaways from the data (in lieu of pricing information). We found that when we compiled the ratings and found the average probability distribution per product that two topics stood out. Topic 5 stood out as the topic that was most represented in the category of rating 1, ie, it was strongly associated with poorly reviewed products. Topic 5 was on the other end of the spectrum in that it was the most popular topic within the high ratings.

What can be gained here is that it seems that topic 5 is associated with problems with printer paper, cartridges and ink. We could infer that these are products within the office category in which people have the most trouble with as installing and setting up a printer can be frustrating. Printers also have a subscription model (ink dries frequently or is used quickly) that customers do not find appealing.

On the other hand, we can see topic 4 has words such as price, use, well, easy and label. Easy to use products at a decent price seem to be a strong recipe for a well reviewed product. Although this is a sample of office products, it seems that these are sensible characteristics that people would look for. A strong rating and positive feedback within a whole set of reviews for a product, can make an additional compelling case for a customer to be nudged into a purchase.

From the seller's perspective, a short term goal could be to alter the product description to highlight the positive elements learned from the product reviews, ie 'easy to use'. In the long term, a consistent, data driven feedback loop on their product's description can be generated via topic modeling.



While working on the office products dataset, we also wanted to try to contextualize our insights amongst other datasets. In other words, were we getting meaningful scores and topics if we performed similar analysis? This was made possible since our code was agnostic to office products, outside specific words we chose to remove via word cloud analysis and examination.

In the same way as office products, we found that there was also a heavy bias towards positive reviews. We performed the same set of analysis including text pre-processing, initial clustering and then more specifically diving into topic modeling via LDA and LSA. We found the most success working with the Arts, Crafts & Sewing dataset. This was more manageable at 500k reviews, which also meant that when we sampled to 100k reviews, we were able to capture more variation; i.e. a greater percentage of the total dataset.

The most interesting findings were also through LDA as we could find a larger set of topics with a higher coherence score than we had with the other data sets. This could possibly indicate there were a broader set of distinctiveness in reviews to understand. This information builds towards the earlier objective “Does the product I am looking for match the characteristics I think are important?” It also points to further surfacing this information to seller’s individual products based on feedback at a high level beyond simply a low or high score on its own.

This success inspired further exploration with an LDA bi-gram (two words make up the document term matrix and corpus) and tri-gram (same as bi-gram, but an arrangement of three words) model. This was to explore what we might find as well as further indications of popular words to look for when understanding the contents of topics.

References

- https://www.datacamp.com/community/tutorials/discovering-hidden-topics-python
- https://nijianmo.github.io/amazon/index.html#sample-metadata
- Jianmo Ni, Jiacheng Li, Julian McAuley Empirical Methods in Natural Language Processing (EMNLP), 2019
- https://www.machinelearningplus.com/nlp/topic-modeling-python-sklearn-examples/
- https://towardsdatascience.com/the-complete-guide-for-topics-extraction-in-python-a6aaa6cedbbc
- https://www.researchgate.net/profile/Diego-Buenano-Fernandez/publication/339368709/figure/fig1/AS:860489982689280@1582168207260/Schematic-of-LDA-algorithm.png