Amazon Feature Mining

Christine Gao, Jia Huang, Peiyang Wen

Overview

Amazon is the leading ecommerce market for various products. The popularity of the site and the diversity of the products means that many consumers trust reviews to help them choose their product.

Often, reviewers will detail certain aspects of each product that they feel **satisfaction/dissatisfaction** toward, in addition to sentiment towards the product (i.e. emotional words like “love” or “happy”). Sentiment analysis can give insight into how people feel toward **various features of their product** and **where they can improve upon**.

Research Question: ***Based on customer reviews, can we determine the most popular product features from Amazon customer reviews, and of these features which are the strongest indicators of customer sentiment about a product?***

Introduction to dataset

Dataset: Amazon Review Dataset (https://nijianmo.github.io/amazon/index.html)

The dataset contains the reviews data and product metadata from May 1996 to Oct 2018. In our project, we choose the fashion category and also run the codes on appliance category as a comparison.

Amazon Review DataSet

- **Total number of reviews:** Fashion Category: 883,636 reviews
- **Variables of interest:**
  - ***reviewText*** - text of each customer review
  - ***overall*** - user ratings of the product (1-5 stars)
  - ***vote*** - *number of other customers who upvoted the review as 'helpful'*

Data Preprocessing

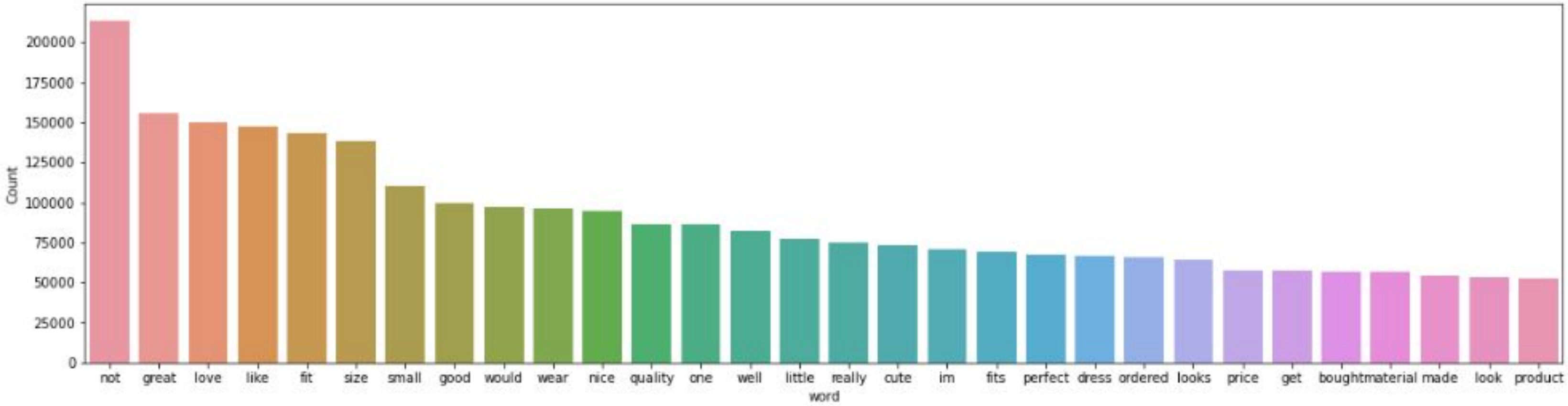
Cleaning and Text preparation (normalization):

1. Converting review text to lowercase
2. Removing punctuation, numbers, and special characters
3. Removing stopwords which do not add meaning to sentences (i.e. ***‘I’, ‘the’, ‘but’, ‘if’, ‘or’***)
   a. Note: We kept ‘not’ in our review text corpus, since it adds sentiment value to a review.
      i. Ex: “I did not like the product” → “I did like product” (which completely changes the meaning of the review)

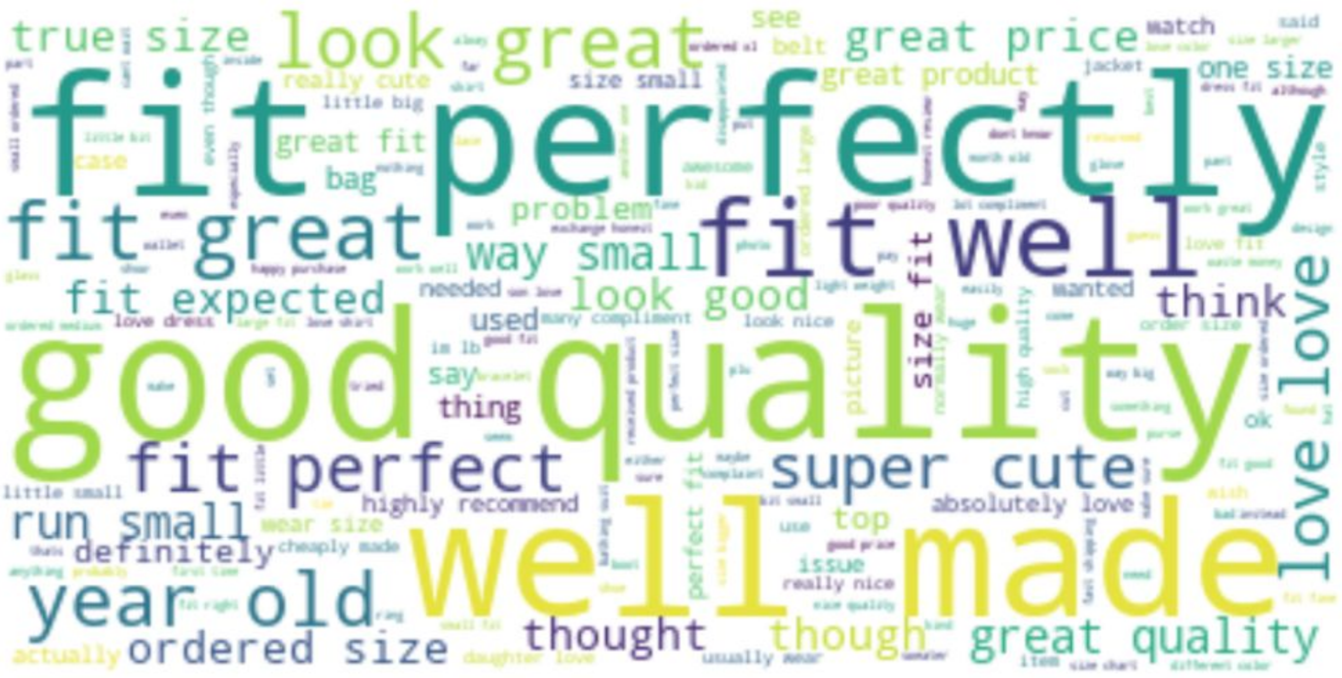
Word frequency visualization

Word frequency after normalizing review text

```
freq_words(df['noStopword'])
```



Word cloud visualization



NLP Methodology

- Feature extraction
- Sentiment analysis
- Topic modeling
- Feature Importance
- Trend Analysis

Tokenization, Lemmatization, and Parts of Speech Tagging (POS)

Tokenization - splitting data into constituent parts

Lemmatization - reducing words to root form

Ex: running, runs, ran ➔ run

POS Tagging - Categorising words to extract meaningful phrases from text (verbs, adverbs, adjectives, nouns, etc.

reviewText	text_clean	noStopword	token	token_lemma	clean	token_pos
I agree with the other review, the opening is too small. I almost bent the hook on some very expensive earrings trying to get these up higher than just the end so they're not seen. Would not buy...	i agree with the other review the opening is too small i almost bent the hook on some very expensive earrings trying to get these up higher than just the end so theyre not seen would not buy aga...	agree review opening small almost bent hook expensive earrings trying get higher end theyre not seen would not buy price not sending back	[agree, review, opening, small, almost, bent, hook, expensive, earrings, trying, get, higher, end, theyre, not, seen, would, not, buy, price, not, sending, back]	[agree, review, opening, small, almost, bent, hook, expensive, earring, trying, get, higher, end, theyre, not, seen, would, not, buy, price, not, sending, back]	agree review opening small almost bent hook expensive earring trying get higher end theyre not seen would not buy price not sending back	[(agree, JJ), (review, NN), (opening, VBG), (small, JJ), (almost, RB), (bent, JJ), (hook, NN), (expensive, JJ), (earrings, NNS), (trying, VBG), (get, VB), (higher, JJR), (end, NN), (theyre, NN), (...

Feature Extraction - Identifying collocations

Collocation - ‘combination of multiple words that have idiosyncratic properties - collection of words that co-occur unusually often’ *(Oxford Dictionary)*

- Ex: ‘machine learning’, ‘fast food’, ‘rich and powerful’

Feature Extraction

Bigram vs Trigram Comparison

```
(pd.Series(nltk.ngrams(word_list, 2)).value_counts())[:20]
```

```
(look, like)        20182
(good, quality)     18743
(fit, perfectly)    17742
(well, made)        16068
(fit, well)         15990
(fit, great)        15290
(look, great)       13214
(year, old)         12799
(like, picture)     12320
(fit, perfect)      11126
(super, cute)       10504
(love, love)        10122
(great, quality)    10096
(run, small)         9762
(way, small)         9141
(really, like)       8658
(fit, like)          8613
(fit, expected)      7968
(look, good)         7843
(great, price)       7774
```

```
(pd.Series(nltk.ngrams(word_list, 3)).value_counts())[:20]
```

```
(look, like, picture)              4232
(love, love, love)                 3073
(nothing, like, picture)           2312
(honest, unbiased, review)         2228
(would, not, recommend)            2185
(exchange, honest, review)         1947
(exchange, honest, unbiased)       1842
(would, definitely, recommend)     1802
(fit, true, size)                  1794
(discount, exchange, honest)       1790
(look, nothing, like)              1777
(im, not, sure)                    1756
(cant, wait, wear)                 1731
(look, exactly, like)              1640
(exactly, like, picture)           1614
(one, size, fit)                   1603
(year, old, daughter)              1542
(received, many, compliment)       1481
(not, look, like)                  1448
(dont, waste, money)               1439
```

Bigram vs Trigram (omitting opinion words)

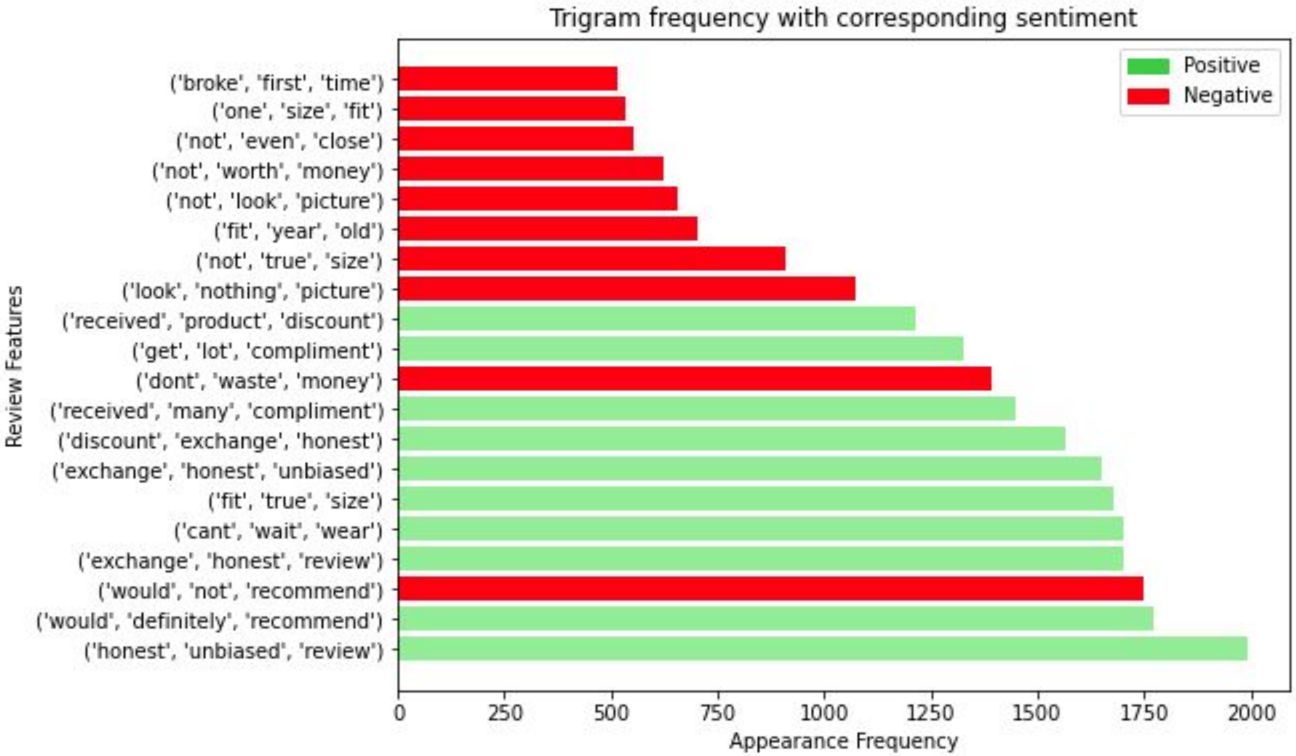
```
(pd.Series(nltk.ngrams(no_opinion, 2)).value_counts())[:20]
```

```
(good, quality)        18848
(fit, perfectly)       17747
(fit, well)            16077
(well, made)           16069
(year, old)            12801
(fit, perfect)         11299
(super, cute)          10506
(run, small)            9768
(way, small)            9142
(look, good)            8207
(fit, expected)         8126
(ordered, size)         7781
(true, size)            7654
(size, fit)             7570
(would, not)            7288
(would, recommend)      7214
(one, size)             6412
(perfect, fit)          6245
(im, not)               6139
(highly, recommend)     6135
```

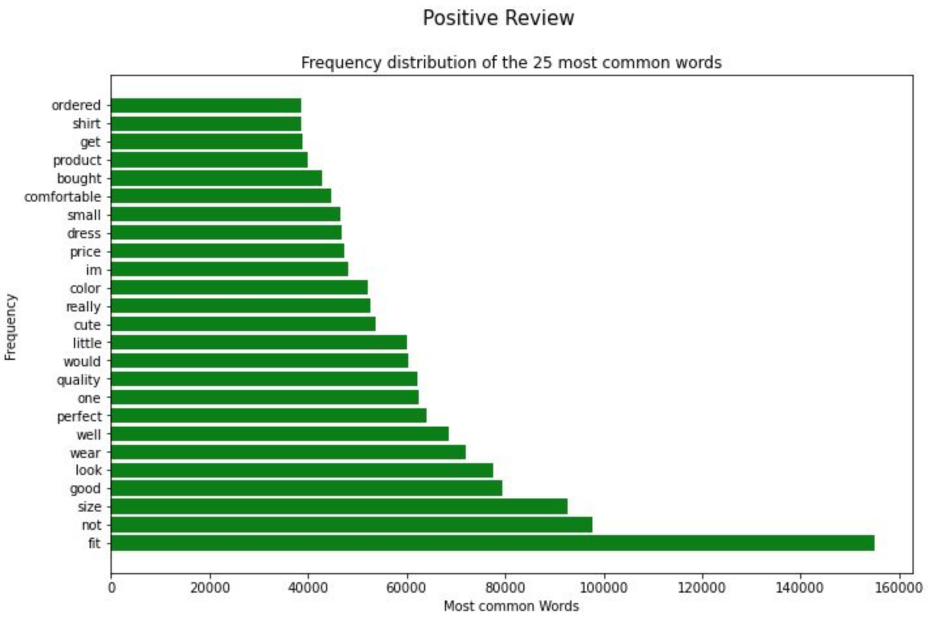
```
(pd.Series(nltk.ngrams(no_opinion, 3)).value_counts())[:20]
```

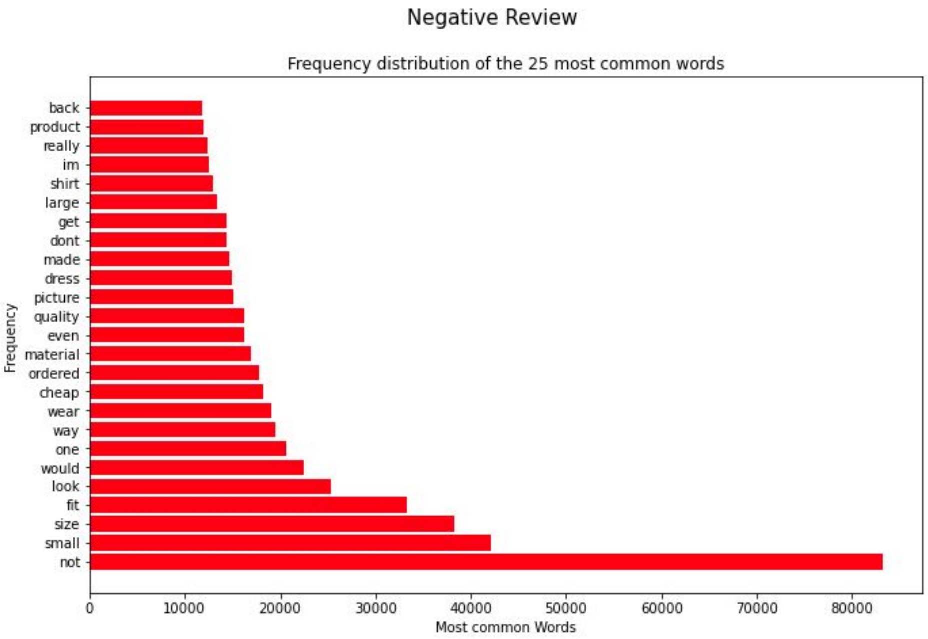
```
(honest, unbiased, review)          2228
(would, not, recommend)             2185
(exchange, honest, review)          1947
(fit, true, size)                   1854
(exchange, honest, unbiased)        1842
(would, definitely, recommend)      1803
(discount, exchange, honest)        1790
(im, not, sure)                     1756
(cant, wait, wear)                  1732
(one, size, fit)                    1614
(year, old, daughter)               1547
(received, many, compliment)        1491
(dont, waste, money)                1442
(fit, year, old)                    1428
(received, product, discount)       1419
(get, lot, compliment)              1397
(run, little, small)                1350
(got, lot, compliment)              1307
(would, highly, recommend)          1237
(not, true, size)                   1223
```

Sentiment Analysis of Feature

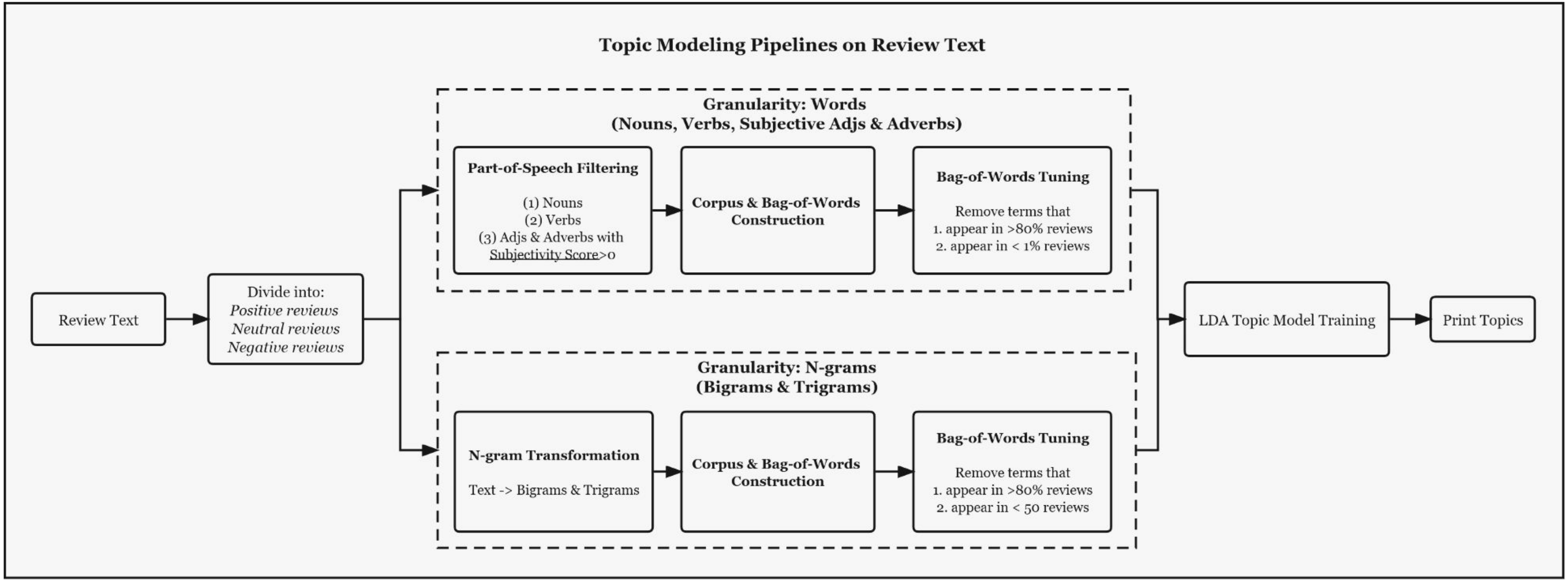


Sentiment Analysis (cont)





Topic Modeling



Topic Modeling - Results

Granularity: Words

Positive Reviews (n_topic=10, num_words=5)

```
[(0, '0.193*"size" + 0.119*"order" + 0.103*"fit" + 0.078*"large" + 0.066*"expect"'), (1, '0.105*"big" + 0.082*"pretty" + 0.079*"little" + 0.069*"perfectly" + 0.065*"hon
est"'), (2, '0.083*"comfortable" + 0.052*"right" + 0.040*"design" + 0.040*"fabric" + 0.037*"want"'), (3, '0.160*"love" + 0.096*"dress" + 0.078*"perfect" + 0.077*"fit" +
0.056*"great"'), (4, '0.040*"top" + 0.037*"get" + 0.030*"wear" + 0.022*"use" + 0.022*"discount"'), (5, '0.090*"wear" + 0.073*"beautiful" + 0.054*"long" + 0.044*"time" +
0.043*"definitely"'), (6, '0.123*"great" + 0.097*"product" + 0.063*"price" + 0.051*"quality" + 0.049*"love"'), (7, '0.150*"good" + 0.096*"nice" + 0.091*"shirt" + 0.085
*"look" + 0.066*"quality"'), (8, '0.078*"color" + 0.058*"love" + 0.056*"soft" + 0.031*"comfortable" + 0.031*"great"'), (9, '0.084*"cute" + 0.059*"review" + 0.052*"smal
l" + 0.039*"receive" + 0.036*"super"')]
```

Neutral Reviews (n_topic=10, num_words=5)

```
[(0, '0.102*"pretty" + 0.082*"fabric" + 0.072*"see" + 0.046*"bad" + 0.036*"kind"'), (1, '0.041*"time" + 0.034*"come" + 0.033*"love" + 0.033*"make" + 0.030*"wear"'), (2,
'0.096*"tight" + 0.080*"give" + 0.042*"area" + 0.042*"wear" + 0.037*"enough"'), (3, '0.205*"top" + 0.076*"keep" + 0.066*"work" + 0.061*"fine" + 0.056*"side"'), (4, '0.1
85*"nice" + 0.100*"bite" + 0.099*"make" + 0.083*"really" + 0.047*"look"'), (5, '0.048*"say" + 0.038*"sure" + 0.035*"use" + 0.035*"know" + 0.033*"old"'), (6, '0.182*"siz
e" + 0.163*"small" + 0.097*"order" + 0.079*"large" + 0.064*"fit"'), (7, '0.066*"dress" + 0.046*"way" + 0.040*"fit" + 0.035*"return" + 0.032*"cute"'), (8, '0.090*"shirt"
+ 0.072*"cute" + 0.063*"good" + 0.054*"material" + 0.054*"fit"'), (9, '0.073*"color" + 0.073*"look" + 0.058*"picture" + 0.037*"cheap" + 0.033*"good"')]
```

Negative Reviews (n_topic=10, num_words=5)

```
[(0, '0.171*"size" + 0.171*"small" + 0.078*"order" + 0.074*"fit" + 0.062*"way"'), (1, '0.162*"look" + 0.086*"picture" + 0.044*"really" + 0.031*"good" + 0.030*"show"'),
(2, '0.114*"give" + 0.068*"nothing" + 0.066*"star" + 0.046*"worth" + 0.038*"price"'), (3, '0.092*"top" + 0.071*"color" + 0.033*"think" + 0.030*"dont" + 0.029*"get"'),
(4, '0.103*"dress" + 0.092*"material" + 0.086*"cheap" + 0.065*"make" + 0.043*"thin"'), (5, '0.129*"shirt" + 0.049*"fit" + 0.047*"love" + 0.043*"tight" + 0.038*"nice"'),
(6, '0.066*"wear" + 0.066*"time" + 0.058*"break" + 0.043*"get" + 0.043*"first"'), (7, '0.093*"quality" + 0.066*"product" + 0.065*"cute" + 0.050*"return" + 0.049*"purcha
se"'), (8, '0.037*"work" + 0.035*"fit" + 0.034*"didnt" + 0.030*"make" + 0.029*"old"'), (9, '0.158*"come" + 0.127*"short" + 0.063*"say" + 0.041*"loose" + 0.030*"skir
t"')]
```

Topic Modeling - Results

Granularity: N-grams (more structured user opinion)

Positive Reviews (n_topic=10, num_words=5)

```
Topic 0  0.015*"fit perfectly" + 0.010*"good quality" + 0.009*"order size" + 0.007*"Very cute" + 0.007*"bath suit" + 0.006*"true size"
Topic 1  0.016*"love them" + 0.012*"little small" + 0.012*"Good quality" + 0.011*"Great quality" + 0.009*"usually wear" + 0.009*"absolutely love"
Topic 2  0.020*"well make" + 0.017*"look great" + 0.015*"honest review" + 0.013*"fit perfect" + 0.009*"fit expect" + 0.009*"little big"
Topic 3: 0.013*"unbiased review" + 0.011*"honest unbiased review" + 0.008*"lot compliment" + 0.008*"exchange honest unbiased" + 0.008*"large size" + 0.007*"get lot"
Topic 4: 0.017*"fit well" + 0.014*"look like" + 0.014*"receive product" + 0.011*"The color" + 0.011*"product discount" + 0.010*"year old"
Topic 5: 0.020*"The material" + 0.015*"Very nice" + 0.014*"honest unbiased" + 0.011*"look good" + 0.010*"great quality" + 0.009*"would recommend"
Topic 6: 0.016*"daughter love" + 0.014*"super cute" + 0.012*"Very comfortable" + 0.010*"wear size" + 0.009*"feel like" + 0.007*"soft comfortable"
Topic 7: 0.018*"fit great" + 0.016*"exchange honest" + 0.014*"really like" + 0.010*"love dress" + 0.009*"Super cute" + 0.009*"The fabric"
```

Neutral Reviews (n_topic=10, num_words=5)

```
Topic 0: 0.012*"The bottom" + 0.010*"fit perfectly" + 0.009*"fit fine" + 0.009*"big size" + 0.009*"The fit" + 0.008*"This shirt"
Topic 1: 0.010*"fit expect" + 0.009*"they 're" + 0.008*"tight around" + 0.007*"wear medium" + 0.007*"bite tight" + 0.006*"Did n't"
Topic 2: 0.023*"order size" + 0.017*"size small" + 0.013*"size large" + 0.011*"size big" + 0.010*"look good" + 0.007*"two size"
Topic 3: 0.015*"fit like" + 0.010*"really cute" + 0.010*"exchange honest" + 0.010*"wear size" + 0.010*"Very cute" + 0.009*"little big"
Topic 4: 0.040*"run small" + 0.015*"order large" + 0.014*"The fabric" + 0.009*"order medium" + 0.008*"material thin" + 0.006*"Very pretty"
Topic 5: 0.010*"large size" + 0.007*"year old" + 0.007*"one size" + 0.007*"normally wear" + 0.007*"bite small" + 0.007*"small expect"
Topic 6: 0.018*"look like" + 0.012*"The material" + 0.009*"way small" + 0.009*"fit well" + 0.008*"like picture" + 0.008*"size chart"
Topic 7: 0.015*"The dress" + 0.014*"good quality" + 0.011*"The top" + 0.011*"usually wear" + 0.010*"little small" + 0.010*"get pay"
```

Negative Reviews (n_topic=10, num_words=5)

```
Topic 0: 0.026*"size chart" + 0.023*"way big" + 0.021*"looked like" + 0.017*"usually wear" + 0.015*"thought would" + 0.013*"cheap looking"
Topic 1: 0.043*"poor quality" + 0.035*"not buy" + 0.020*"not like" + 0.018*"send back" + 0.016*"even though" + 0.015*"can not"
Topic 2: 0.073*"look like" + 0.017*"dress not" + 0.016*"not happy" + 0.015*"definitely not" + 0.014*"true size" + 0.014*"not true"
Topic 3: 0.019*"feel like" + 0.018*"still small" + 0.017*"fit well" + 0.016*"material thin" + 0.016*"material not" + 0.013*"extremely small"
Topic 4: 0.042*"like picture" + 0.033*"nothing like" + 0.030*"waste money" + 0.028*"year old" + 0.023*"ordered size" + 0.018*"nothing like picture"
Topic 5: 0.026*"cheaply made" + 0.026*"not even" + 0.023*"size small" + 0.022*"not worth" + 0.021*"would not" + 0.021*"not recommend"
Topic 6: 0.069*"way small" + 0.034*"not fit" + 0.019*"small ordered" + 0.017*"poorly made" + 0.016*"im not" + 0.016*"bathing suit"
Topic 7: 0.045*"fit like" + 0.037*"run small" + 0.022*"not good" + 0.020*"ordered xl" + 0.017*"small not" + 0.016*"normally wear"
```

Topic Modeling - Discoveries

Positive Reviews

Pros:

1. **Material** is comfortable.
2. **Size** fit perfectly.
3. **Style** is very cute.
4. **Discount** is good.
5. Worth the **price**.

Good in all aspects.

Neutral Reviews

Pros:

1. Great **quality**
2. **Size** fit (for some)
3. **Look** good

Cons:

1. **Size** for most clothes not fitting well (due to online shopping).

*Generally good except **size**.*

Negative Reviews

Cons:

1. **Size** not fitting well.
2. **Quality** is bad.
3. **Not like Picture**.
4. **Waste money**.

Bad in various ways.

Other Review Features

	rating	vote	length	prop_of_noun	prop_of_verb	prop_of_adj_adverb	subjectivity	readability
0	5.0	0.0	3.0	0.33	0.33	0.33	0.25	97.02
1	2.0	3.0	23.0	0.26	0.35	0.26	0.43	65.53
2	4.0	0.0	27.0	0.33	0.26	0.19	0.38	57.15
3	5.0	0.0	3.0	0.00	0.33	0.67	0.25	97.02
4	4.0	0.0	18.0	0.39	0.28	0.22	0.50	69.81

Comments:

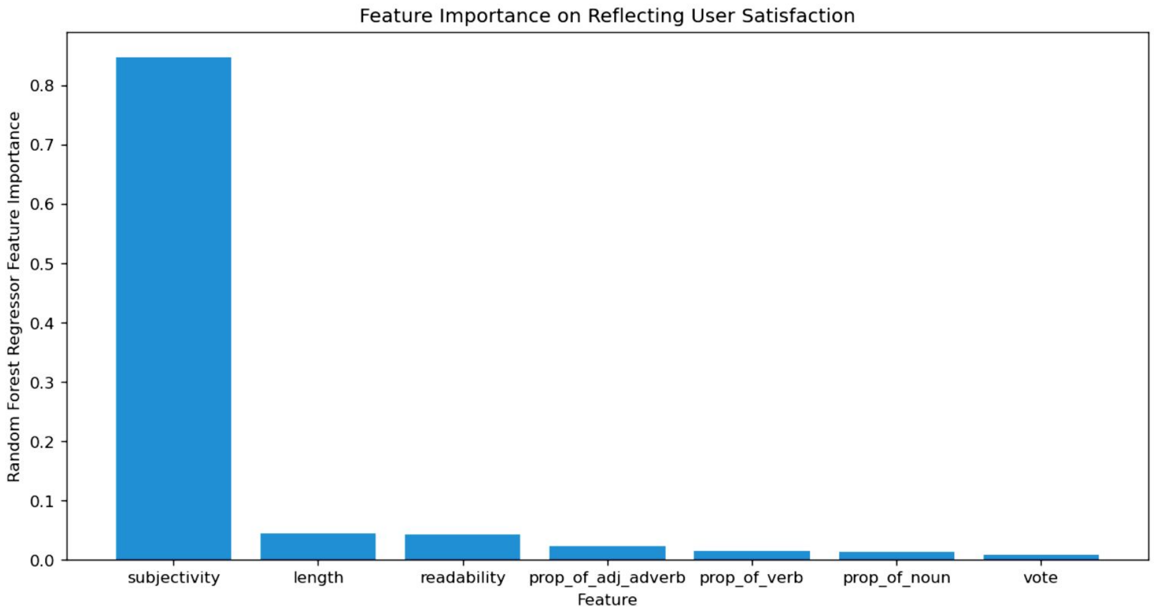
1. **Subjectivity:** Calculated using the *TextBlob.sentiment.subjectivity* package. This score quantifies the amount of personal opinion and factual information contained in the text.

2. **Readability:** Our method uses the Flesch Reading Ease (FRE) score to measure how readable the review is:

$$206.835 - 1.015 \left(\frac{\text{total words}}{\text{total sentences}} \right) - 84.6 \left(\frac{\text{total syllables}}{\text{total words}} \right)$$

Feature Importance

Q1: Which feature best reflects user satisfaction on the product?



Model: Random Forest Regressor

Y Feature: Rating
X Feature: Other Features

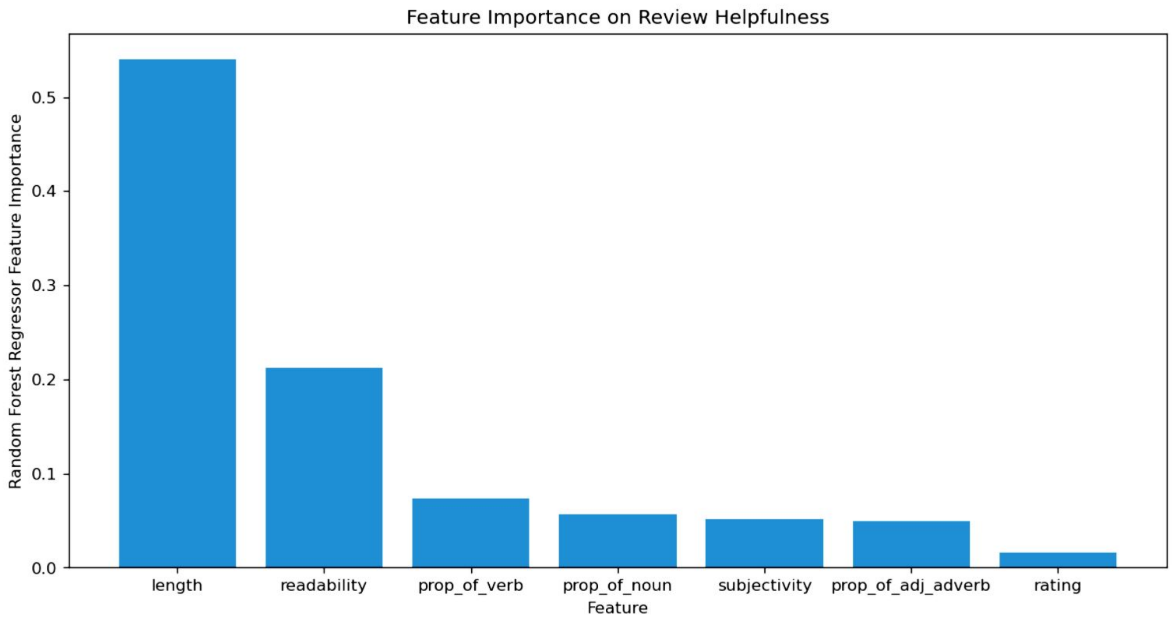
Hyperparams Tuning: GridSearchCV

Observations:

1. Users with high satisfaction tended to give **subjective** reviews with higher **prop of adjs** used.
2. Users with high satisfaction tended to write **longer** reviews with good **readability**.

Feature Importance

Q2: Which features best influences review helpfulness?



Model: Random Forest Regressor

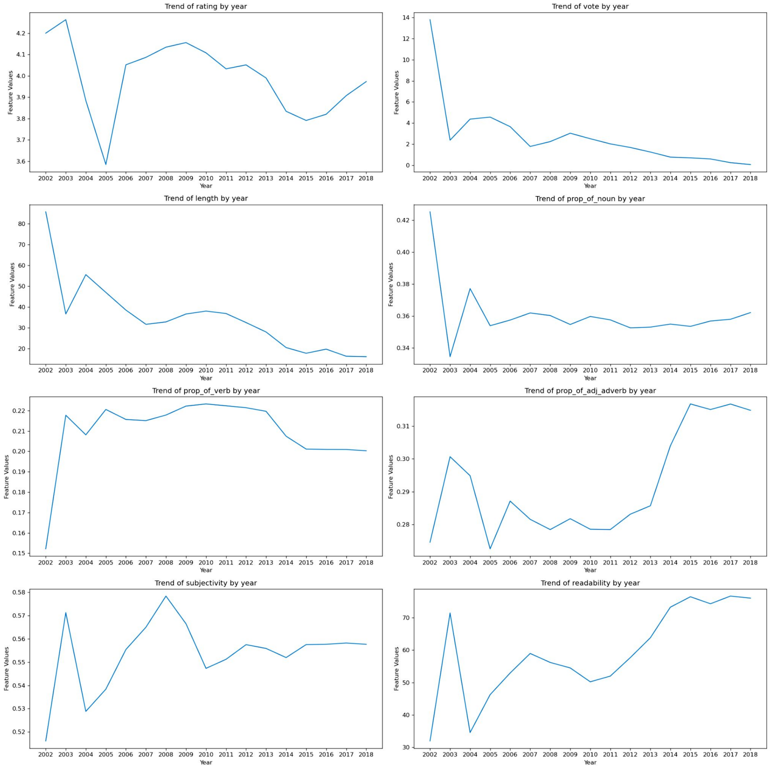
Y Feature: Helpfulness votes
X Feature: Other Features

Hyperparams Tuning: GridSearchCV

Observations:

1. **Long** reviews with g**ood readability**, **subjective opinions** and **higher prop of nouns** (product-related features) are more likely to be endorsed by readers due to richer information.

Trend Analysis - Review Features



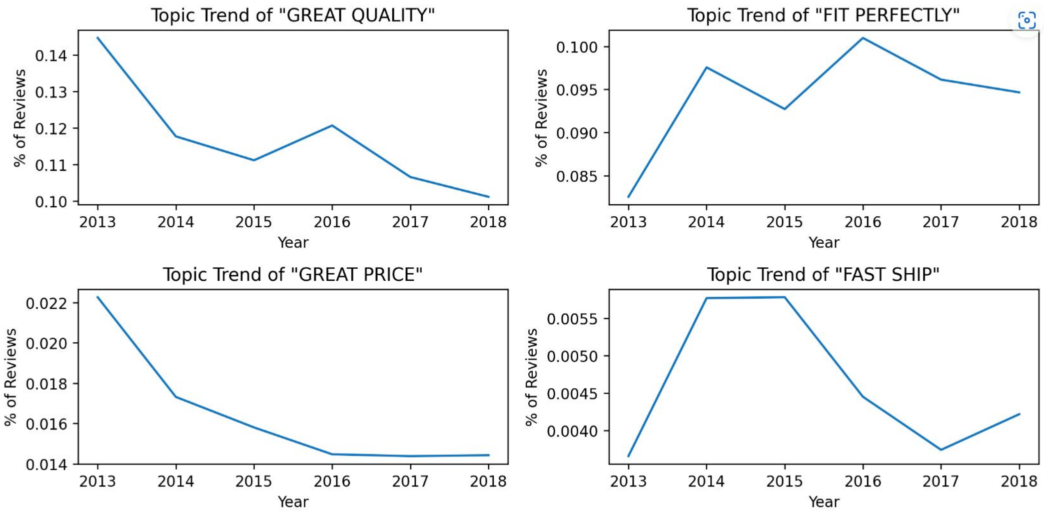
Observations:

1. Downward Trend:
    a. Helpfulness Votes (publish time)
    b. Average Review Length

2. Upward Trend:
    a. Prop of Adjs & Adverbs
    b. Average Readability (due to length)

In Fashion category, consumers tended to express more personal and subjective experiences. With shorter reviews on average, the readability increased accordingly.

Trend Analysis - Topics



Methodology:

1. **Metric:** *% of reviews containing the topic-related keywords*.
2. **Time Period:** 2013-2018.
3. **Unit:** Bigrams.
4. **Synonym Matching:** Bigrams with similar meanings are matched to one topic.

Observations:

1. Customers' satisfaction on fashion product quality and price was decreasing.
2. The size picking difficulty during online shopping was getting optimized.

Conclusion

- **From Review text:**

Word frequencies, feature extraction, and topic modeling show similar results:

Customers mainly care about the fit, quality and price of the products in the fashion category. The majority of complaints show dissatisfaction with these features. Size picking stands out as a major challenge for online shopping.

- **Other information included:** (ratings, length of reviews, votes, readability, etc.)

The feature importance calculation with random forest regression shows:

1. **Feature that best reflects the satisfaction level: subjectivity.**
   Customers tend to use strong emotional words and personal stories to express their satisfactions toward products.

2. **Feature that best indicates the review helpfulness level: length of reviews.**
   Potential buyers prefer reviews with richer information, easy-to-read content, stronger opinions and more discussions on product features.

Appendix

Project: Amazon Fashion Review Analysis

Dataset: Amazon Review Data

Thank You