1- Methodology

The data chosen consist of different types of data from nominal to the text format and also some non-English characters. In order to extract the useful information out of the data, Python development environment has been used to deduce useful results in the form of different graphs patterns and visuals. The current case study of unsupervised learning consists of five major phases. Phase one includes data fetching in the form of Python data frame. Phase two concerns initial preprocessing of data in which data is filtered from irrelevant information and STOPWORDS. Phase three focuses on LDA topic modeling. Phase four deals with the implementation of NLTK's VADER sentiment analyzer as well as classification of positive, negative & neutral reviews and representation of these on cloud vis. Phase five describes aspect opinion and positive/negative word clouds.

After the declaration of libraries, a namespace is created to make sure that all the names in a program are unique and can be used without any conflict. We have used local namespace type under the function name flag. Input file "productReviewShopee_1.csv" has been loaded into a data frame of dimension (11416, 6). The data frame contains columns with names title, rating, date, categorie, comments, product_option. For the sake of sentiment analysis, our focus is on "comments" column. Initially the random and unordered data has to be made in a form where it can be used. The cleaning of data has been done and the anomalies have been removed from the data. By looking insights to the data following tasks has been done in cleaning the data and making it in a usable form:

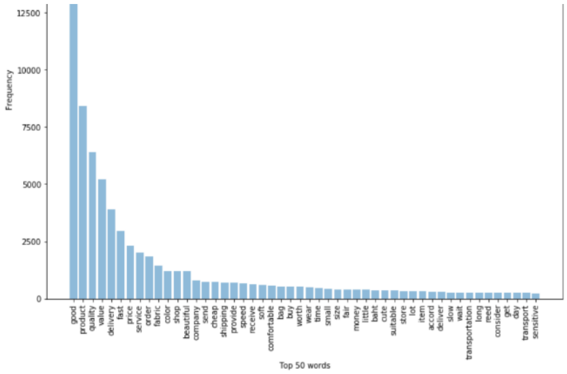
- Handling missing values
- Removing punctuation marks
- Removing stopwords
- Creating new column

It has been found that there are 14 missing comments (NaN) in the data, and rows count drops to 11402 with 6 features (columns). Further, punctuation marks have been removed from comments data using Regular Expression Operations (re). For further filtration of data, we tokenized each comment in the data frame. Tokenization is a process in which a document or sentence break down into words, phrases or other meaningful elements called tokens. After tokenization, stopwords have been removed from comments. Stopwords are those words that usually occur frequently in documents and are not needed for sentiment analysis. To remove the stopwords, first we have defined stopwords as 'like', 'etc', articles (a, an, the) and added other English language stopwords predefined. The reason to clean our data is that the cleaner the data, the more suitable they are for mining and feature extraction, which leads to the improved accuracy of results. After removing all tokens that belong to stopwords; punctuation, blank spaces, numeric characters etc, lemmatization of each token has been done and a new data frame column "comments_lemma" is created. Lemmatization is the process of converting a word to its base form by using NLP spaCy's lemmatizer: parse the sentence using the loaded model object nlp and extract the lemma for each token. Examples of lemmatization:

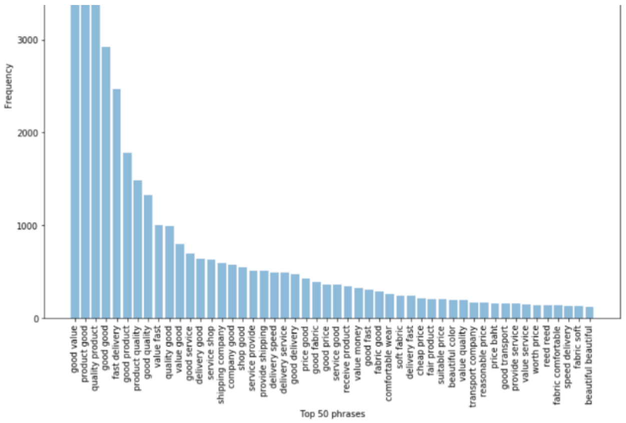
- "accessories" to "accessory"
- "receiving" to "receive"
- "forgot" to "forget"

When preprocessing steps of phase two are complete, the dataset is ready for sentiment analysis and tags classification.

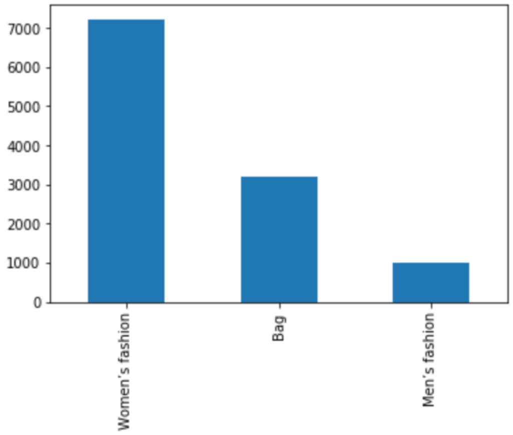
In vectorizing the data, each text should be represented as a vector with length equal to the vocabulary size. Each dimension of this vector corresponds to the occurrence of a word in a text. CountVectorizer from scikit-learn is used to convert a raw text definition to a matrix of token counts. After applying vectorization, we got 11207 number of total samples and 808 number of total features. Ngram-range is the value that decides top occurring words or top occurring phrases. Here is the list of top fifty most frequent words in comments.



”good” is the most frequent word in comments that has been reported about 17500 times. Similarly, by changing ngram value to (2,2), we got top fifty phrases in comments. This is very interesting that why we need to figure out phrases when we have top fifty frequent words? Sometimes single word gives us positive sentiment by applying sentiment analyzer however actual sentiment of sentence or phrase is negative. For example, if we pass a sentence like “The car is not good” to analyzer, it may give positive sentiment by giving importance to word “good” however if we go on phrases instead of words, our accuracy of sentiment analysis will be enhanced. As discussed, top fifty phrases are shown below in a bar graph created by matplotlib.pyplot.



Bar plot of three categories mentioned is also shown below. It is interesting to know that women’s fashion has maximum comments.



For feature engineering and topic modeling, we have used LDA instead of TF-IDF. Latent Dirichlet Allocation (LDA) unsupervised algorithm is a generative probabilistic model. This model is used to uncover hidden structure in a collection of text. LDA is a generative probabilistic model that assumes each topic is a mixture over an underlying set of words, and each document is a mixture of over a set of topic probabilities. We can describe the generative process of LDA as, given the M number of documents, N number of words, and prior K number of topics, the model trains to output:

(i) psi, the distribution of words for each topic

(ii) Kphi, the distribution of topics for each document i

We have used psi, and there are five topics (clusters) derived in the code with 10 words or phrases with each topic. These words or phrases are the most relevant or frequent ones in each topic. 5 topics of whole data are measured and topics by category are also measured. Topics are plotted LDA topic modeling visualization using pyLDAvis library. Definitions of visual elements in LDA visualization:

Circles

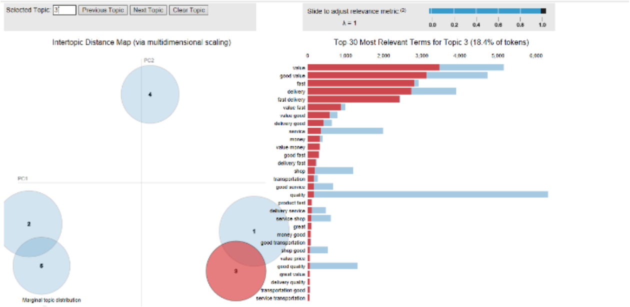
pyLDAvis displays N circles. N is number of categories. Each circle represents each category whose areas are set to be proportional to the proportions of the categories across the total tokens in the text (definition). The area of circle for topic is set to be proportional to the estimated number of tokens that were generated by topic across all definitions. The Jensen-Shannon divergence is used to compute distances between topics. The Jensen–Shannon divergence is a method of measuring the similarity between two probability distributions.

Red Bars

Red horizontal bars represent the estimated number of times a given term was generated by a given category. When a category(topic) is selected, the red bars are shown for the 30 most relevant terms for the selected topic.

Blue Bars

Blue horizontal bars represent the overall frequency of each term in all definitions.

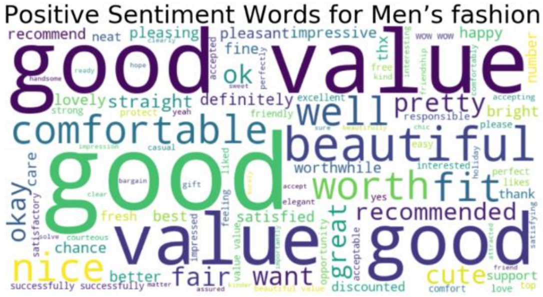


In the above picture, top 30 most relevant terms of third topic are evaluated. Blue bar shows the total frequency of the particular term where as red bar shows frequency in the third topic. Topic visualizations for women's fashion, bag and men's fashion have also been created separately.

Now data is all set for both, by category wise and overall comments sentiment analysis. In order to calculate and analyze the weight of sentiment for each individual word, VADER sentiment analysis algorithm is used to understand the sentiment power of words which are then learnt by the model to predict the testing examples. VADER is a rule-based sentiment analysis tool and a lexicon that is used to express sentiments in social media. VADER uses a combination of A sentiment lexicon is a list of lexical features (e.g., words) which are generally labelled according to their semantic orientation as either positive or negative. First, we created a sentiment intensity analyzer to categorize our dataset. Then four functions have been created to get compound score, positive/neutral/negative tag and functions for positive and negative words. The compound value is a useful metric for measuring the sentiment in a given comment. In the proposed method, the threshold values used to categorize comments as positive, negative or neutral are:

Positive sentiment: compound value >= 0.05, assign tag “Positive”
Neutral sentiment: compound value > -0.05, assign tag “Neutral”
else
Negative sentiment: (compound value <= -0.05), assign tag “Negative”

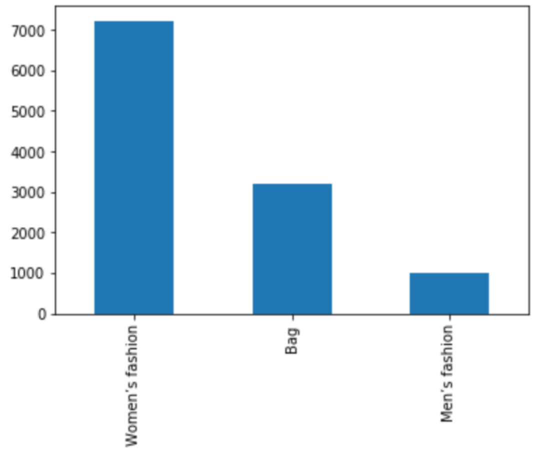
Four columns of data frame, “sent_score”, “sentiment”, “positive_words”, “negative_words” for compound score, sentiment tag, positive words and negative words have been created. With the help of “word_cloud_vis”, overall positive and negative words are displayed along with their frequency weight.



Positive & negative words for each individual category (women’s fashion, bag, men’s fashion) have been displayed by “word_cloud_vis”.
Aspect extraction is a fundamental task of opinion mining or sentiment analysis. It aims to extract opinion targets from opinion text. For example, from “This phone has a good screen,” it aims to extract “screen.” In product reviews, aspects are product attributes or features. They are needed in many sentiment analysis applications. The opinion mining is done and saved in the form a csv file.

2- Results:

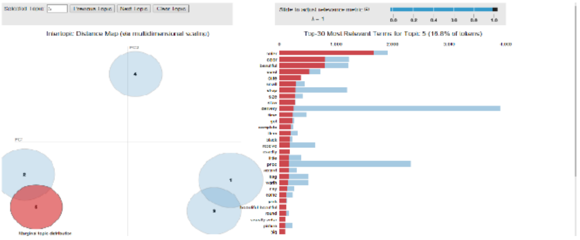
One thing needs to be cleared in this particular problem is that problem statement can be addressed in general or by categories. For this, we need to understand that sentiment analysis has been done on general data as well as on each category (bag, women’s fashion, men’s fashion). Here is bar graph of categories.



Pay special attention to the distribution of reviews by category, maximum number of reviews have been reported on women’s fashion and only 10% approximately, reviews are reported for men’s fashion. Its quite clear from the distribution that reviews on women’s fashion will dominate while conducting sentiment analysis of all reviews generally.

LDA has been used to find top fifty words and phrases. TF-IDF should not be used with LDA algorithm because LDA is a probabilistic model that tries to estimate probability distributions for topics in documents and words in topics. Re-weighting the TF's by its IDF's would disproportionally increase the chance of rare words being sampled, making them have a stronger influence in topic assignment.

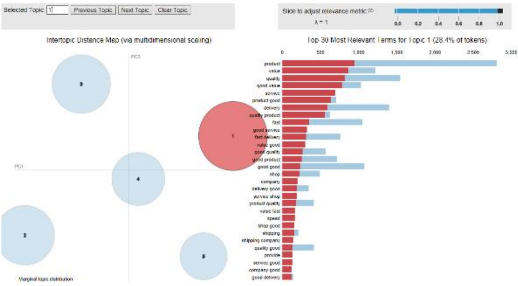
Topic modeling is done with five topics and 10 words in each topic. For each topic, top 30 relevant terms have been labelled in red bar while blue bar shows total frequency of the particular term. Since topics are clusters of words, we can monitor each topic relevant words in contrast with overall relevance of that particular term. This analysis shows us how close each term relevance is with that of overall relevance against relevance metric lambda.



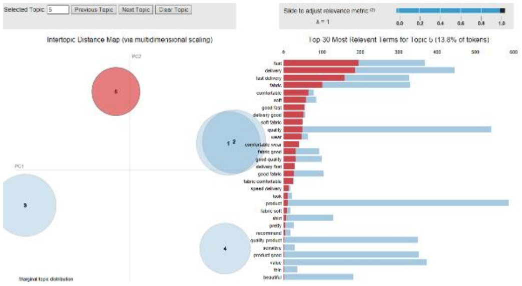
Topics Modeling for Overall Comments

Also in the graph, one can view that topics 1&3, 2&5 have overlapping of topics terms that means both 1&3, 2&5 pair share some of its terms in common.

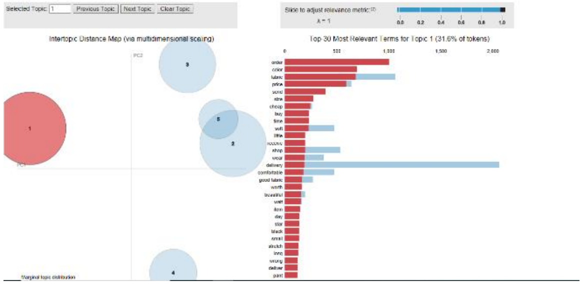
Apart from topics wise overall terms plotting, one might be interested in category wise topic's terms representation. For this topics terms for each category has been visualized.



Topics Modeling for Bags

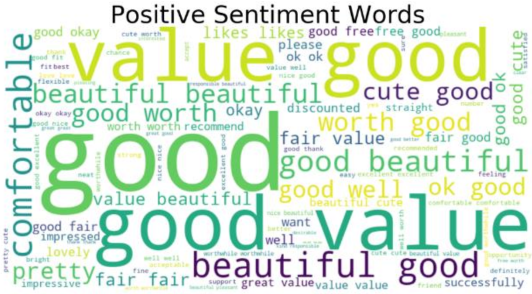


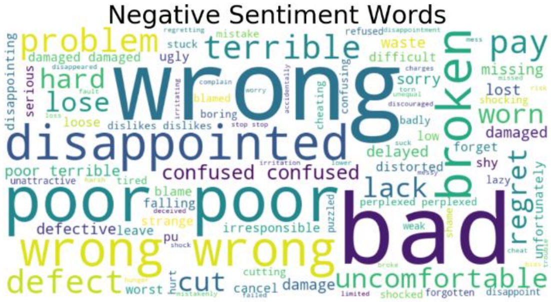
Topics Modeling for Men's Fashion



Topics Modeling for Women's Fashion

VADER sentiment analysis gives top 100 positive & negative words after implementation on comments. This verifies the sentiment analysis as positive & negative words can be assessed after sentiment taging on basis of compound score. This model can be used as unsupervised learning model and can be used to predict further social textual data.





Word Distribution and Word Cloud: This section discussed the prominent words in the textual data. By visualizing the words in word cloud, it's easy to see how customers use the language about the products and which words are most positive.

The word clouds show the words that have been used the most from the customers in their reviews, which gives their interests towards a particular product. But on the other hand, there also exists the flaws of these world clouds, that they only are showing the distribution of the individual words. Which have the high probability of removing the context of the word, and meanwhile it also take no notice of negative prefixes. Further on, n-grams will be used in order to solve this problem. N-grams increase the size of the values to be observed from one word, and the counts to be conducted to word sequences.

3- Conclusions:

With the advancement of technological change, business processes specially e-commerce has been shifted from manual to internet-based solutions in order to increase the recurrent revenue. This paper focuses on the analysis of sales data of an e-commerce store in order to identify the hidden patterns of purchase history and to predict the scenarios where user preferred to recommend or not recommend the products. We found that, women's fashion is most reviewed and popular. By applying unsupervised learning algorithms on textual data (Reviews), we found that, the words like good, value good, beautiful good, comfortable, good value etc. when used in reviews, the products had positive reviews. It turns out that, prediction model technique can be used to have the highest accuracy and precision value in predicting the whether the product is liked or not liked on the basis of customers reviews.