**Assignment 2**

The purpose of this assignment is to extract question sentences and imperative sentences from review texts. Specifically, the question sentences should contain question mask '?' and adjective phrases. Similarly, the imperative sentences should contain exclamation mark '!', adjective phrases, and should start with verbs.

Before extracting the required sentences, one must import the NLTK package and other necessary functions. Then we should extract the review text from review Content document and store that to review import text.

A screenshot of a cell phone

Description automatically generated

By using the len function, we find that the review\_text contains 88129772 words.

A close up of a logo

Description automatically generated

Now, since we want to extract sentences from the review text, we need to parse those texts to sentences by using RegExr. Before that, we should replace '\n' with blank space and then use RegExr to parse sentence.

A close up of a logo

Description automatically generated

After parsing the review text, we have 1202086 sentences.

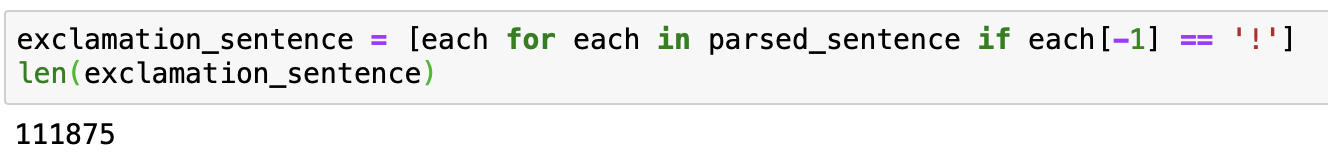


## Sentence Extraction Part

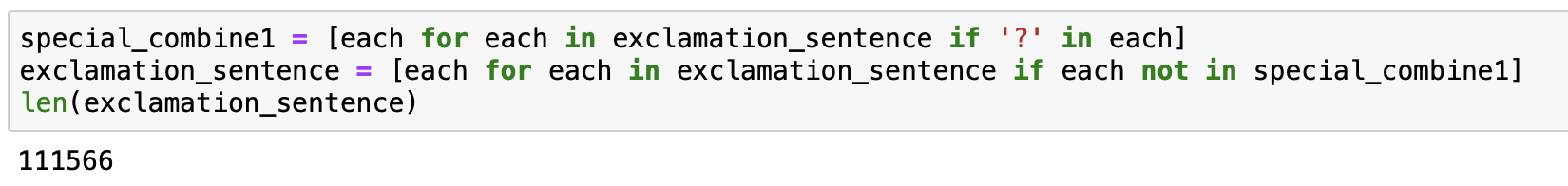
To get a collection of imperative sentences that satiate requirements mentioned above, we need to split those pared sentences into three which ends with exclamation mark, question mark, and stop mark respectively.

### **Exclamation Sentence**

### For sentence that stops with exclamation mark, there are 111875 of them.



However, after carefully review those sentences, we might find that some of them also contain question mark(s), making it difficult to identify those sentences. Then we need to find those sentences and store them into the special\_combine1 object. Next, we might need to eliminate those sentences from the original exclamation sentence. Finally, we can find that the number of exclamation sentence has reduced to 111566.



### **Question Sentence**

Similarly, executing the same procedures in the processes of collecting question sentences and normal sentences which end with stop mark.

A screenshot of a cell phone

Description automatically generated

Find and store sentence that contains both '!' and '?' to special\_combine2. Eliminate those sentences from the original question\_sentence.

A screenshot of a cell phone

Description automatically generated

Combining the two special\_combine list to one. We find that the number of sentences that contains both '?' and '!' is 433.

A screenshot of a cell phone

Description automatically generated

Checking those special sentences, we find that the emotion of those sentence is similar with that of exclamation sentence. Then we decide to combine the special\_combine with exclamation\_sentence.

A screenshot of a cell phone

Description automatically generated

### **Declarative Sentence**

### Finally, we can get a collection of sentences that ends with stop sign ('.'). For those sentences, we find that some of them contain '!' or '?' inside. Then we can extract and save those sentences to special\_combine3 (that contains '!') and to special\_combine4 (that contain '?') respectively. Likewise, we should also eliminate those sentences from the original stop\_sentence.

A screenshot of a cell phone

Description automatically generated

Additionally, if we go through the special\_combine3 and special\_combine4 separately, we find that the emotion of those sentence is similar with that of exclamation\_sentence while that of special\_combine4 is similar with that of question sentence(?).

A screenshot of a social media post

Description automatically generated

## POS Tagging and CFG Analysis

Gathering those sentences is not the end, we need to go further to find sentence that has 'adjective phrases' which might require use to use the POS tagging techniques.   
  
**Question Sentence POS Tagging**

* First of all, we might focus on the question sentence processing.
* After tokenizing and tagging the question sentences, we can use 'JJ', 'JJR', and 'JJS' to find if those sentences contain these three tags.
* If so, adding the tagged sentence to question\_sentence\_JJ object.

A screenshot of a cell phone

Description automatically generated

Now, we can claim that the number of question sentence that contain 'adjective phrases' is 4174.



Given that we might need the content of these tagged sentences, we have to transform these tagged sentences to their original format (which are without tags). At last, storing the result to question\_sentences object.

A picture containing knife

Description automatically generated

**Exclamation Sentence POS Tagging and CFG analysis**   
To extract imperative sentences from the collected exclamation sentences, we need two steps.

* Firstly, we need to find sentences that contain 'Adjective Phrases'.
* Secondly, we need to find sentences that start with verb. For the first step, we can use the function of RegexpParser to define the tag of the first verb. Here, we have to consider that some sentence might start with adv.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

We only get 2658 sentences that satisfy the requirments of having '!' and 'Adjective Phrases' and of starting with verb.



Let's convert and store the sentence to a list object. The final result will be stored to imperative sentences object.

A screenshot of a cell phone

Description automatically generated

### **Statistics Summary**

The data frame listed below shows statistics information. Specifically, we can summarize that the number of required question sentence is 4174 and that of required imperative sentence is 2125. Comparing the number of sentences with and without adjective phrases, the required question sentence (with adjective phrases) is half of that of original question sentences. The number of required imperative sentences is one sixth of that of original imperative sentences.

A screenshot of a cell phone

Description automatically generated

## Analysis Part

### In this section, statistics calculation, bigram analysis and trigram analysis will be conducted.

### **For Question\_sentence Analysis**

Calculate the average words per sentence and the number of unique words contain in the question sentences.

A screenshot of a cell phone

Description automatically generated

**Unigram Analysis:**   
Before removing the punctuations and stopwords, one can get the top 20 tokens that have the top frequency.

A screenshot of a cell phone

Description automatically generated

A close up of a keyboard

Description automatically generated

The top 20 tokens do not provide much meaningful information since most of them are punctuations and stopwords. We can find that '?' has appeared 4742 times because what we are processing is the collection of question sentences. Besides, most of them are degerminators or prepositions. Hence, we need to eliminate those words or punctuations.

A screenshot of a cell phone

Description automatically generated

A close up of electronics

Description automatically generated

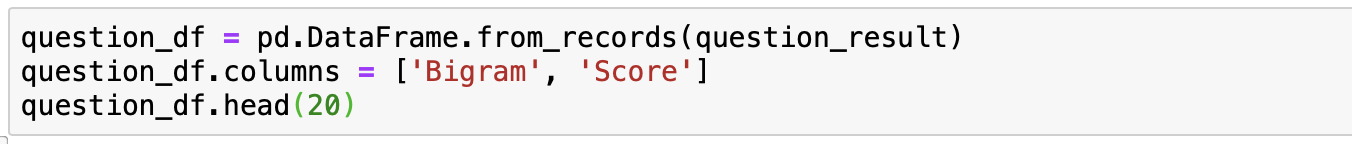
According to the result, after excluding several function words like 'Would', 'Like', 'Maybe', and 'Could' etc., we can find that the ninth word is **'Good'**, indicating that the attitude of customers toward these reviewed products might be positive. Normally, the word **'right'** indicates 'positive attitude'. However, given that we are analyzing the emotion of a whole sentences, we cannot draw an conclusion right now.   
Besides, words like **'wear'**, **'fit'**, **'price'** also suggest that customers care about the product quality or product price. The appearance of these words in a collection of question sentences might suggest that people either complained issues that related with those words or they used rhetorical questions for being thought-provoking.   
However, without checking the constituency of words in a sentence, we cannot make any conclusion. Therefore, it is necessary to conduct a Bigram analysis.

**Bigram Analysis:**   
Now we can use the build-in lemmatize to identify the root of each tokens. The reason for lemmatization is that there are bigrams that are same but will be treated as different pairs, such as “Looks Good” and “Look Good”.  
Then, by using the BigramCollocationFinder and the score\_ngrams function, we can get pairs of bigrams as well as their corresponding score.   
Now, **after the process of bigram splitting**, we get bigrams that might contain punctuation and stop word inside. Since we want the results to be meaningful, we need to use filter to **exclude those non-alphabetic tokens and to eliminate pairs that contain any stop word**.

Finally, we will get the result.

A screenshot of a cell phone

Description automatically generated



A screenshot of a cell phone

Description automatically generated

Compared with the result we obtained from the unigram analysis which only focuses on singular word, we can confirm the emotion of the collection of question sentences now.

After the analysis of the top 30 bigrams, we can make a conclusion that, generally, customers hold a positive attitude toward these products since 8 out of the top 30 bigrams are positive. Specifically, those positive bigrams are **('look', 'great'), ('look', 'good'), ('well', 'made'), ('great', 'price'), ('good', 'quality'), and ('high', 'quality')** etc. If we simply focus on these bigrams, it is obvious that the price is reasonable, the quality if great, and the products look great.    
In most cases, people might use question sentence to expresses their confusion, surprise, or even angry. For interpreting the existence of these positive words under this circumstance, we can make an assumption that customers might use rhetorical question to compliment products. Even for the appearance of bigram **('go', 'wrong')**, customers might write down comment, such as "How could you go wrong for getting that product?", to expresses their confirmation for the purchased products.

### **For Imperative Sentence Analysis**

### Calculate the average words per sentence and the the number of unique words contain in the question sentences.

A screenshot of a cell phone

Description automatically generated

**Unigram Analysis:**   
Before removing the punctuations and stopwords, one can get the top 20 tokens that have the top frequency.

A screenshot of a cell phone

Description automatically generated

A close up of a keyboard

Description automatically generated

Again, these tokens do not provide much information. Hence, eliminating stop words and punctuations are necessary.

A screenshot of a cell phone

Description automatically generated

A close up of a keyboard

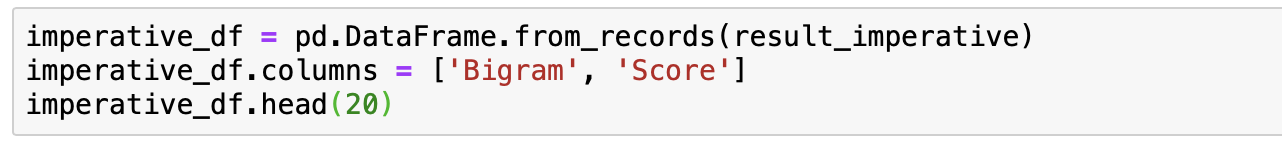
Description automatically generated

Given the nature of imperative sentences, verbs that displayed above indicate the actions the reviewers had taken or show the emotion of the reviewers. Sometimes, they also reflect the attitude of customers. Take the verb \*\*'love'\*\* as an example, ‘love’ has the highest word frequency in the list, suggesting that many reviewers have used ‘love’ to describe their feeling. The word with the second highest frequency is ‘great’. By combing those words together, one can speculate that reviewers “think the products they purchased are great and they love them”. However, we cannot make a conclusion in this stage. Similarly, the next one is bigram analysis.

**Bigram Analysis:**   
Again, we can use the build-in lemmatizer to identify the root of each tokens. Next, we can get pairs of bigrams as well as their corresponding score by using BigramCollocationFinder. Then, we need to use filter to exclude those non-alphabetic tokens and to eliminate pairs that contain any stop word. Finally, we will get the result.

A screenshot of a cell phone

Description automatically generated



A screenshot of a cell phone

Description automatically generated

After analyzing of the top 20 bigrams, we can make a conclusion that, generally, customers still hold a positive attitude. Specifically, the score of bigrams **(looks, great)** is approximately four times than that of bigram **(works, great),** reflecting that 'looks great' has a relatively high frequency. Besides, despite the top 2 bigrams, other positive bigrams such as **(fits, great), (great, price), (worked, great), (good, quality), and (look, great)** etc. are also included in the list. Besides, the top 5 bigrams are all positive which already indicate that the collection of imperative sentences are quite positive. Even the reminder of the bigrams, most of them are positive. The conclusion we draw in the bigram analysis is same as the conclusion of unigram analysis.

Sentiment Analysis

The reason why we should conduct a sentiment analysis is that we can only use unigrams or bigrams to speculate the customer altitude. It is impossible to get a numeric indicator to reflect the degree of satisfaction or degree of dissatisfaction. By using the sentiment intensity analyzer embedded in the nltk package.

### **Imperative Sentences Analysis:**

Using the polarity\_scores, we **can vectorize each word listed in the sentences** based on the word of bags that embedded in the SIA function. Then, we can get a numeric result based on the input text (or each sentence). The results are consistent with ‘Positive values’, ‘Negative value’, and ‘Neutral value’. Next, after assigning different weights to the three value, the build-in function will generate a compound value which indicates the direction and degree of satisfaction level.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

Then we should extract the details of ‘compound value’. The maximum is 0.98 while the minimum is -0.87. The mean is 0.45 while the median is 0.61, suggesting that the distribution of those sentences is skewed.

A screenshot of a cell phone

Description automatically generated

To better identify each sentence, we can create a new column named label which store numeric variable. For the sentence that is being labeled as '1', it implies that this sentence should be positive. If it is being labeled as '0', it is neutral. If the label is '-1', then it is negative. In this case, we set the threshold as '0.45' and '-0.45', which is the mean we obtained from the “compound value”.

Then, we can list 3 sentences from the Positive Sentence Collection (labeled as 1) and the Negative Sentence Collection (labeled as -1).

A screenshot of a cell phone

Description automatically generated

To visualize the results, we can draw a bar plot which illustrates the number of positive, neutral, and negative sentences. We can find that approximately 62 percent of the sentences are positive. Nearly 35 percent of sentences are neutral while the rest of the sentences, which accounts for about 3% of the total number of imperative sentences, are negative. The result confirms the conclusion we obtained from the unigram and bigram analysis.

A screenshot of a cell phone

Description automatically generated

**Question Sentences Analysis:**

For question sentences analysis, the procedures will be exactly the same as what we conducted for the imperative sentences.

A screenshot of a cell phone

Description automatically generated

Now we need to set the thresholds to differentiate the positive, neutral, and negative sentences. By concluding the information obtained from the ‘compound value’, we find the simply using mean as the threshold is inappropriate since the distribution of the collection is extremely right-skewed. Then we decide to set the lower boundary as ‘0’. For sentence whose compound value is below 0, it belongs to negative sentence collection. The value of upper boundary is ‘0.13’, implying that sentence with compound score that is higher than 0.13 will be treated as positive sentence. The rest will belong to the collection of Neutral Sentences.

**A screenshot of a cell phone

Description automatically generated**

Again, we can go thro­­ugh the top 3 sentences from positive sentence collection and negative sentence collection.

A screenshot of a cell phone

Description automatically generated

Next, we can visualize the result. Here, we find that number of positive sentences (41%) is similar with that of neutral sentences (39%). Negative sentences take account for 20%. Now, we can make a conclusion that the reviewers stay a generally positive or neutral attitude when they wrote down the sentences. A few of them hold a negative attitude. This is different compared with the conclusion that we obtained from the unigram and bigram analysis. This could be either that people tend to be neutral (or positive) when they used question sentence to express their felling or it is inappropriate to use SIA to analyze question sentences.

A screenshot of a cell phone

Description automatically generated