OMIS 670
SOCIAL MEDIA ANALYTICS FOR BUSINESS
PROJECT REPORT
SPRING 2021

FLIPKART REVIEWS IN INDIA
GROUP - 12

TEAM MEMBERS

BEHROOZ ALIPOOR
XUEFEN YI
NISHITH VARMA PENUMATSA

TABLE OF CONTENTS

SUMMARY

This project focuses on the Top 9 popular Headphones on Flipkart which is India's largest e-commerce platform. The analysis contains models and different visualizations that help online merchants, headset manufacturers and other related stakeholders understand the Indian market, as well as make business plans and decisions to initiate their business in India.

1. INTRODUCTION

1.1. Industry: E-Commerce

1.2. Business Scenario

Due to the pandemic, more and more people are choosing to shop online. Online shopping has seen explosive growth and because of the requirements of the work from home, the demand for headphones, microphones and other accessories needed for home office has soared.

The current business scenario for this project is to help companies enter e-commerce in India with the analysis and critical consulting for business needs and decision making.

First, we conducted a sentiment analysis which helps merchants to understand the positive or negative reviews that customers have about the product. Next, we performed topic modeling which can find out the reasons and keywords that affect product reviews. This helps merchants redefine their customer's needs and improve their products. This is essential to get more customers in the fierce competition in the market.

1.3. Challenges

Initially, to overcome serious challenges, it is necessary to clean the data using different analytical tools, because either the format of columns are not correct, or records have missing values. Next, we need to pre-set valuable questions, set up suitable models, and perform correct analysis.

2. DATA DESCRIPTION

2.1. Data source

https://www.kaggle.com/naushads/flipkart-reviews

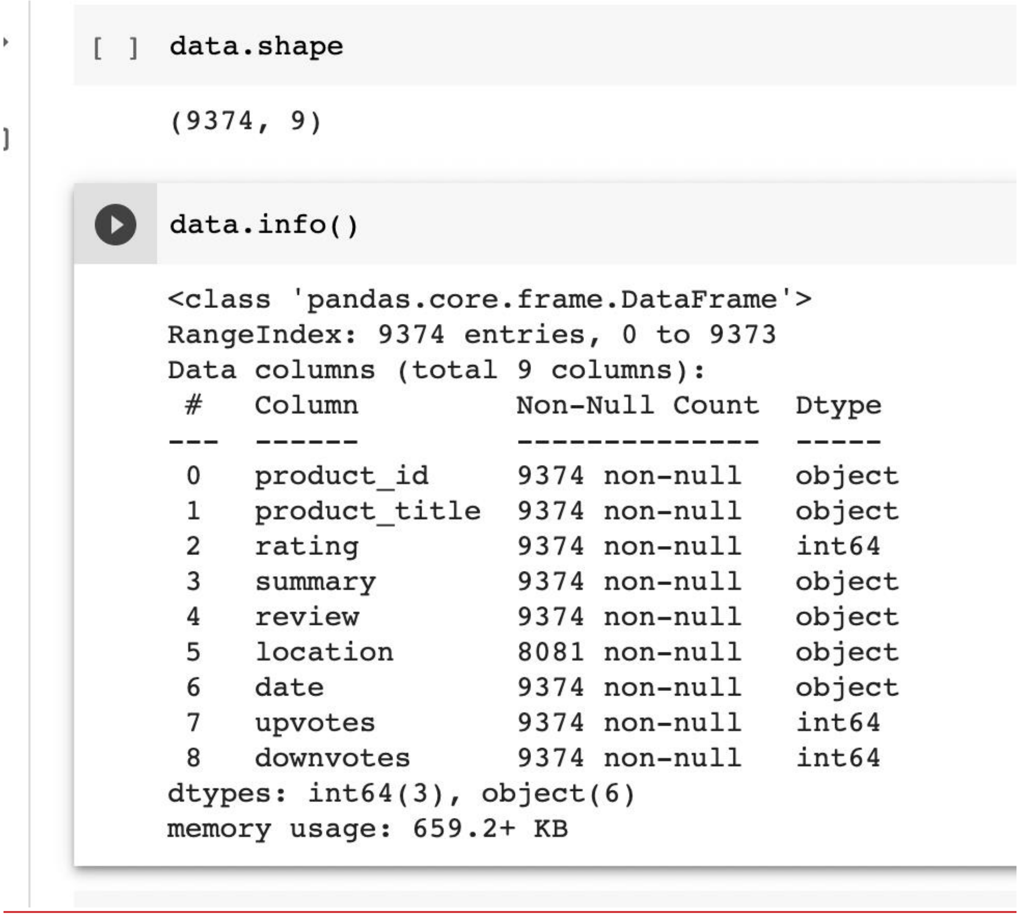
2.2. Data Description

This dataset was obtained by means of web scraping and listed 9374 reviews of Top 9 headphones in Flipkart which includes product_id, product_title, rating, summary, review, location, last_time_reviewed, date, upvotes and downvotes.

Each review is written by the customer and can be used for the vendor’s product purchase plan and the manufacturer's production plan in future.

2.3. Data Pre-processing

The dataset contains 9374 rows and a total of 9 columns as shown in the figure below. We do not have any missing values in our dataset other than the location column with 8081 values. As the location attribute does not affect our analyses, we dropped the attribute and carried forward.



```
[ ]  data.shape
```

```
(9374, 9)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9374 entries, 0 to 9373
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   product_id     9374 non-null   object
 1   product_title  9374 non-null   object
 2   rating         9374 non-null   int64
 3   summary        9374 non-null   object
 4   review         9374 non-null   object
 5   location       8081 non-null   object
 6   date           9374 non-null   object
 7   upvotes        9374 non-null   int64
 8   downvotes      9374 non-null   int64
dtypes: int64(3), object(6)
memory usage: 659.2+ KB
```

Tools and Programming languages:

- Sentiment Analysis: Python
- Topic Modelling: Python
- Visualization: Power BI
- Word Cloud: Python

2.4. Variables

Following variables are the total variables in the Flipkart dataset.

- Product_id
- Product_title
- Rating
- Summary
- Review
- Location
- Date
- last_time_reviewed
- Upvotes
- Downvotes

Following variables are the important variables we used for our analysis:

- Product_title: The name of the headphone product. The headphone types can be identified from the title. (Wired/ Bluetooth)
- Rating: Rating is given on a scale of 1 to 5 given by the customers for the product.
- Summary: The keywords or title of the reviews.
- Review: Actual and detailed reviews about products.
- Location: Location of the customers.
- Date: The date when the review was posted.
- Upvotes: The number of other customers who up-voted this review.
- Downvotes: The number of other customers who downvoted this review.

3. AUDIENCE

The data is analyzed considering the customers and headphone manufacturers to be a stakeholder. We can provide the service to any business and manufacturers who want to start the business on Flipkart.

For the majority of sellers, building a community and a marketing list is an afterthought, so they have to use other techniques. Before you start selling a new product, you should be gathering the information needed during the product research phase. Part of product research should also entail developing a list of influencers, groups, and places to reach your demographic.

Product ratings and reviews secure the customer journey by giving them the confidence to complete the checkout process. User-generated content provides valuable consumer insights. It helps Ecommerce retailers understand the needs of consumers.

4. MODELS

4.1. Sentiment Analysis

Since customers will truly express their feelings and thoughts in product reviews, using sentiment analysis to detect emotions in the data is an important step in understanding customers. Text data can be divided into subjective and objective texts. Objective texts are relatively neutral sentiments. And subjective texts contain more pronounced positive or negative emotions. Based on this sentiment analysis, we analyze the positive and negative reviews with respect to product quality and delivery service. In our approach, firstly tokenization and lemmatization were performed in order to remove the contractions and other unnecessary characters from the reviews using Natural Language Processing (NLP). Later the text was further cleaned, and all the stop words were removed excluding the positive and negative attitude words to perform sentiment analysis. A word cloud of all the words was plotted using the "TextBlob" library in Python. In our analyses, the polarity and their subjectivity were checked for each review which defines if the review has a positive, negative or neutral sentiment accordingly. Further visualizations were plotted for the number of words with each sentiment and for the top words in each sentiment accordingly.

4.2. Topic Modeling:

Topic modeling is the detection of words or phrases in a set of text data and the categorization of similar word expressions. After sentiment analysis, we used topic modeling to extract the most common words in positive and negative evaluations. Identifying those keywords helps Flipkart sellers understand their customers better, so they can sell more popular products and provide better service to meet customers' needs. On the other hand, it can help Flipkart sellers better identify customer dissatisfaction and make quick responses and changes.

Topic Modeling can be divided into two methods: Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA). In this project, we used Latent Dirichlet Allocation (LDA) using Python.

5. ANALYSIS

5.1. Sentiment Analysis

The Figure No.1 shows the total number of ratings by location in Indian cities. There are a total of 2025 cities from where customers rated the products. The cities of Bengaluru, New Delhi, Kolkata, Mumbai, and Hyderabad are the top cities with highest number of ratings, whereas Chhatral, Chennith and Chebrolu are the cities lowest cities with lowest number of ratings.

The cities with the highest number of ratings indicate that these products were sold and extensively used, whereas the cities with lowest number of ratings indicate that these products did not have a considerable number of sales in those regions. Here are some of the top cities with higher ratings.

- Bengaluru, New Delhi, Kolkata, Mumbai, Hyderabad, Chennai, Pune, Patna, Lucknow, Ahmedabad, Thane, Bangalore

Below is the list of all products offered within the cities in India.

- BoAt Rockerz 235v2 with ASAP charging Version 5.0 Bluetooth Headset
- OnePlus Bullets Wireless Z Bluetooth Headset
- BoAt Bass Heads 100 Wired Headset
- Realme Buds 2 Wired Headset
- OnePlus Bullets Wireless Z Bass Edition Bluetooth Headset
- U&I Titanic Series - Low Price Bluetooth Neckband Bluetooth Headset
- BoAt Airdrops 131 Bluetooth Headset
- Realme Buds Q Bluetooth Headset
- Realme Buds Wireless Bluetooth Headset

Figure No.2 shows the products with higher rates by customers. There are 9 types of products in total. The product named BoAt Rockers 235v2 with ASAP charging version 5.0 Bluetooth Headset with almost 8 thousand rates is the highest rated product, whereas the Realme Buds Q Bluetooth Headset with less than 4 thousand rates is the lowest rated product by the customers. It clearly indicates that the highest rated products were purchased and used by customers within different cities in India.

Moreover, based on the sentiment analysis in figure No.3, customers provided 7769 positive, 720 neutral and 885 negative ratings which indicates customers rated the products positively and are satisfied.

Figure No.4 indicates that over 80.91% of customers upvoted the products and over 19.09% of the customers downvoted the products. It clearly shows that most of the customers were satisfied as a result of use and purchase of these products within the cities in India.

Figure No.5 shows that over 38 thousand rates customers provided last time in total. Customers provided a higher number of rates 8 months ago compared to seven months ago or 9 months ago. It denotes that products were either released newly to the markets of these cities or customers were requested to rate the products in the abovementioned cities in India.

5.2. Topic Modeling Analysis

- Figure No.7 not only shows all the reviews by customers within different cities in India, but the top 20 words customers used. Among all these reviews, the words "good", quality", "sound" and "product" are the most used words by the customers.
- Figure No.8 shows the top 10 positive words customers used within different cities in India. Customers provided their positive feedback and review about the 9 products in their cities based on the quality and price of the products. These top 10 words are "good, quality, sound, product, bass, not, very, best, nice, price".
- Figure No.9 shows the top 10 negative words customers used within different cities in India. These words were used to negate the products based on their quality, price and other related topics. These top 10 negative words listed are "not, bad, product, very, quality, work, good, buy, bass and sound".
- Figure No.10 shows the top 10 neutral words customers provided within different cities in India. Customers neither had negative nor positive reviews of the products.

  These top 10 neutral words listed are "not, work, product, buy, one, use, month, osm, properly, and day".

  Comparing all these three figures, we come up with a result that customers mostly liked and interested to purchase and buy these products within the cities in India.

6. FINDINGS

Question#1: What is the highest rated product customers preferred to purchase and why?

There are 9 types of products and based upon analysis on Figure No. 2, we can observe that the product named "BoAt Rockerz 235v2 with ASAP charging Version 5.0 Bluetooth Headset" has been rated the highest product among other products by the customers within different cities in India. This product received almost eight thousand reviews and these reviews are positive, negative and neutral. Based on the sentiment analysis, there are 7769 positive reviews and 885 negative reviews whereas there are 720 neutral reviews provided by the customers. Moreover, we can say that since the number of positive reviews is greater than other reviews, therefore, customers wanted this product due to good quality, price, better performance etc.

In conclusion, as a result of sentiment analysis, the polarity and subjectivity are 0.162 and 0.431 which is between 0 and 1 and indicates a positive result, which shows customers mostly reviewed the product based on its characteristics and specifications.

Figure#5: Number of ratings by last time reviewed represents the total number of product rating at the same time. And we can see that the top two blue squares show that 4,700 ratings and 3850 ratings were completed eight months ago and nine months ago. Secondly, many ratings were completed six or seven months ago. In addition, the third orange square shows that there were 2,620 customers rated a month ago. The customers rated this month has a very significant increase. There are more than 11000 ratings. The most time of reviews is like the time of covid-19 lockdown in India. It is speculated that many people buy headsets because they need to work at home.

Question#2: Which city is a better choice to initiate the business?

As a result of analysis conducted, we observed in Figure No.1 that there are a total of 2025 cities that customers rated the products in India.

Among all, the city of Bengaluru is the top city where most of the rated products have been purchased by the customers within the given period.

Therefore, the city of Bengaluru is recommended for future investment by the entrepreneurs, stakeholders and business owners. Moreover, other cities such as New Delhi and Kolkata are preferred successively.

Question#3: Which business strategy is considered regarding customer reviews in order to compete with other companies in the long run?

Based on the reviews and ratings customers provided for the products, it is necessary for the business to opt a business strategy to differentiate itself from other companies or industries. There are different types of business strategies such as competing with businesses based on price, by using a product with unique features, selection of a small portion of the market, and considering low cost and differentiation from others.

According to the LDA Model in figure no.12, we observed the top four sentences that customers used many times such as “good quality sound bass price” and “product nice delivery money super” as well as number of positive reviews, we can clearly suggest a long term strategy such as considering low cost and differentiation from others based on uniqueness of the product which entail more customers with low cost and uniqueness of the product which itself causes the business to lock the customer for a long period of time in the market.

7. RECOMMENDATIONS

Based on the Finding 1, we believe that Bluetooth headsets are more preferred by customers over wired headsets. Although there are two wired headphones inside the top 9 rating headphones, wireless headphones are significantly more rated. We recommended that businesses should give priority to Bluetooth headphones when purchasing.

Customers have a clear preference for the brand of headphones. The top 9 rated headphones are mainly from the three brands: BoAt, Realme, and OnePlus. We recommended that businesses should consider the brand of the headphones when purchasing, giving priority to BoAt, Realme, and OnePlus's headphones

Also, Due to covid-19, many people need to work from home. Sales of headphones and other supplies needed for home office have increased significantly. We recommend that businesses should consider changes in the Covid-19 pandemic. When people return to work in the office from home, sales of headphones which are suitable for the office will increase.

According to Finding 2, customers buying the headphones are mainly from India's major cities such as Bangalore, New Delhi, and Kolkata. Among them, Bangalore has the highest number of reviews from customers. This demonstrates that major cities such as Bangalore have a large customer demand and market share. Therefore, we recommend businesses to start their business in big cities, especially Bangalore.

On the other hand, we also noted that more than 5,900 reviews locations are blank and did not come from these major cities. This shows that there is still a large portion of customer demand and market share outside of the major cities. Therefore, we suggest that if companies want to expand into new markets and increase headphone sales, outside of major cities markets are the best choice. And entering new markets outside the main cities in advance can also avoid incentive competition.

From the finding No.3, we noticed the key words from Topic Modeling Analysis, quality, sound, bass, delivery, and price. These keywords can be divided into two categories, product characters and price.

First strategy is for Price-sensitive customers. Those price-sensitive consumers are not willing to pay a few extra costs for headphones. Keeping headphones has lower prices is the key to locking in this part of the market. Therefore, we recommend that companies sell a few mid-qualities but low-cost headsets.

Second strategy is for customers who care more about quality than price. These types of customers need good quality headphones and are willing to pay more for it. Our recommendation is that companies should focus on the quality of headphones, develop and sell headphones with better sound, especially bass. These headphones can be priced higher.

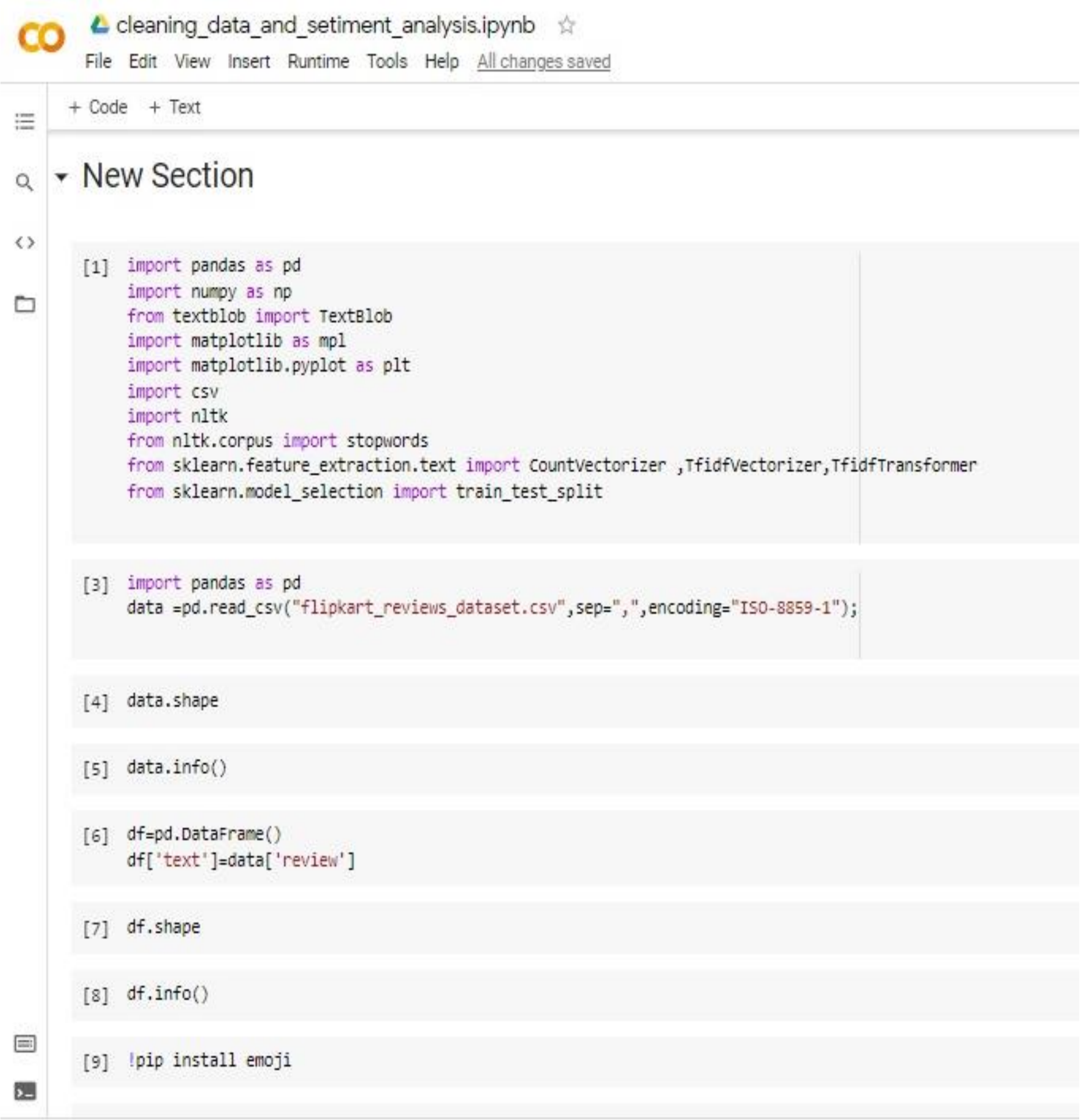
In addition, whether it is a low-price strategy or a high-quality strategy, product delivery is a part that enterprises need to pay attention to. Because when the headphones are similar, all customers will choose the products which can be delivered earlier.

8. CODE SNIPPETS

8.1. Python scripts used for Sentiment Analysis.

8.1.1. Text cleaning using Natural Language Processing

Here the text reviews are cleaned for contractions using lemmatization and the stop words are removed.



```
[1] import pandas as pd
    import numpy as np
    from textblob import TextBlob
    import matplotlib as mpl
    import matplotlib.pyplot as plt
    import csv
    import nltk
    from nltk.corpus import stopwords
    from sklearn.feature_extraction.text import CountVectorizer ,TfidfVectorizer,TfidfTransformer
    from sklearn.model_selection import train_test_split
```

```
[3] import pandas as pd
    data =pd.read_csv("flipkart_reviews_dataset.csv",sep=",",encoding="ISO-8859-1");
```

```
[4] data.shape
```

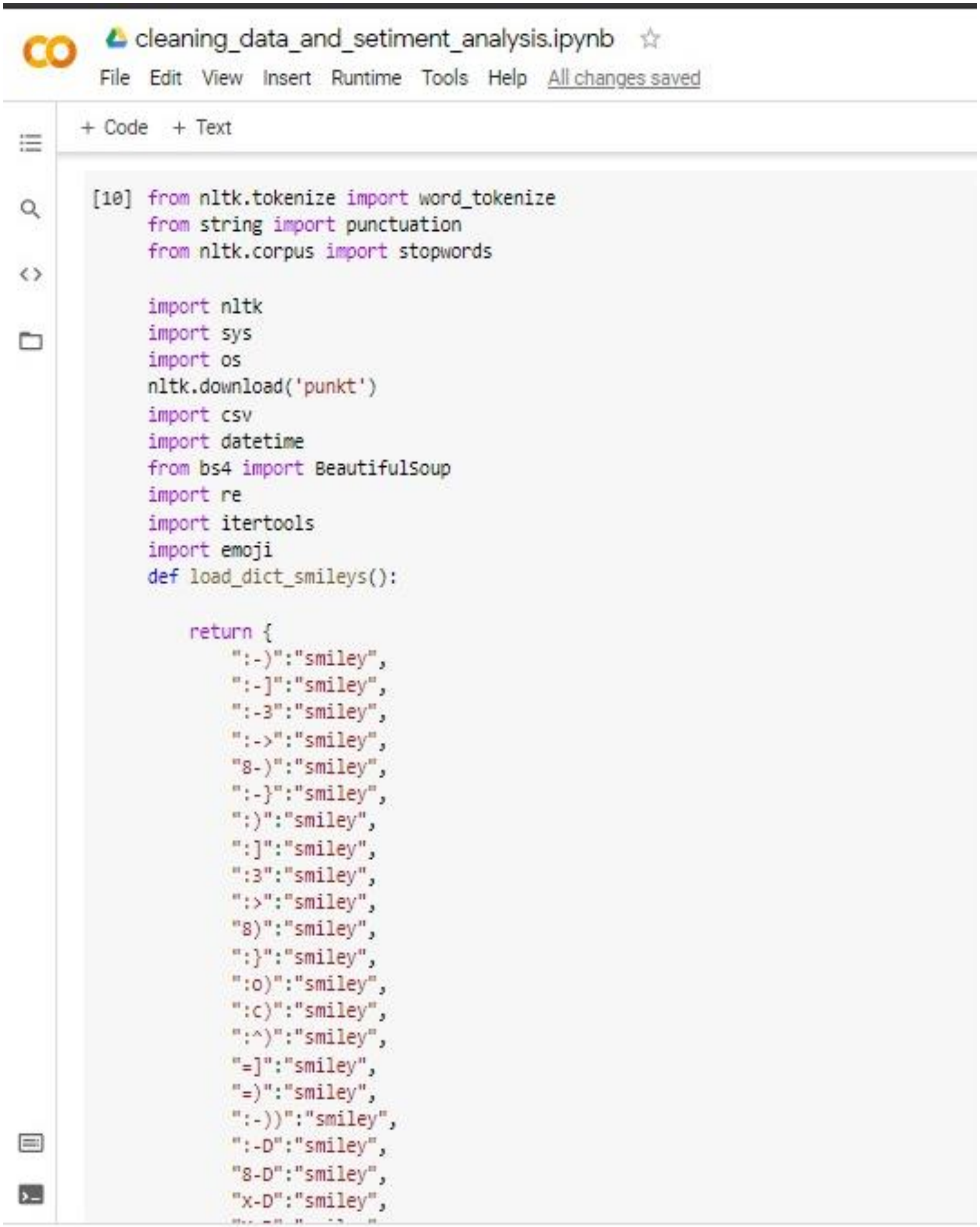
```
[5] data.info()
```

```
[6] df=pd.DataFrame()
    df['text']=data['review']
```

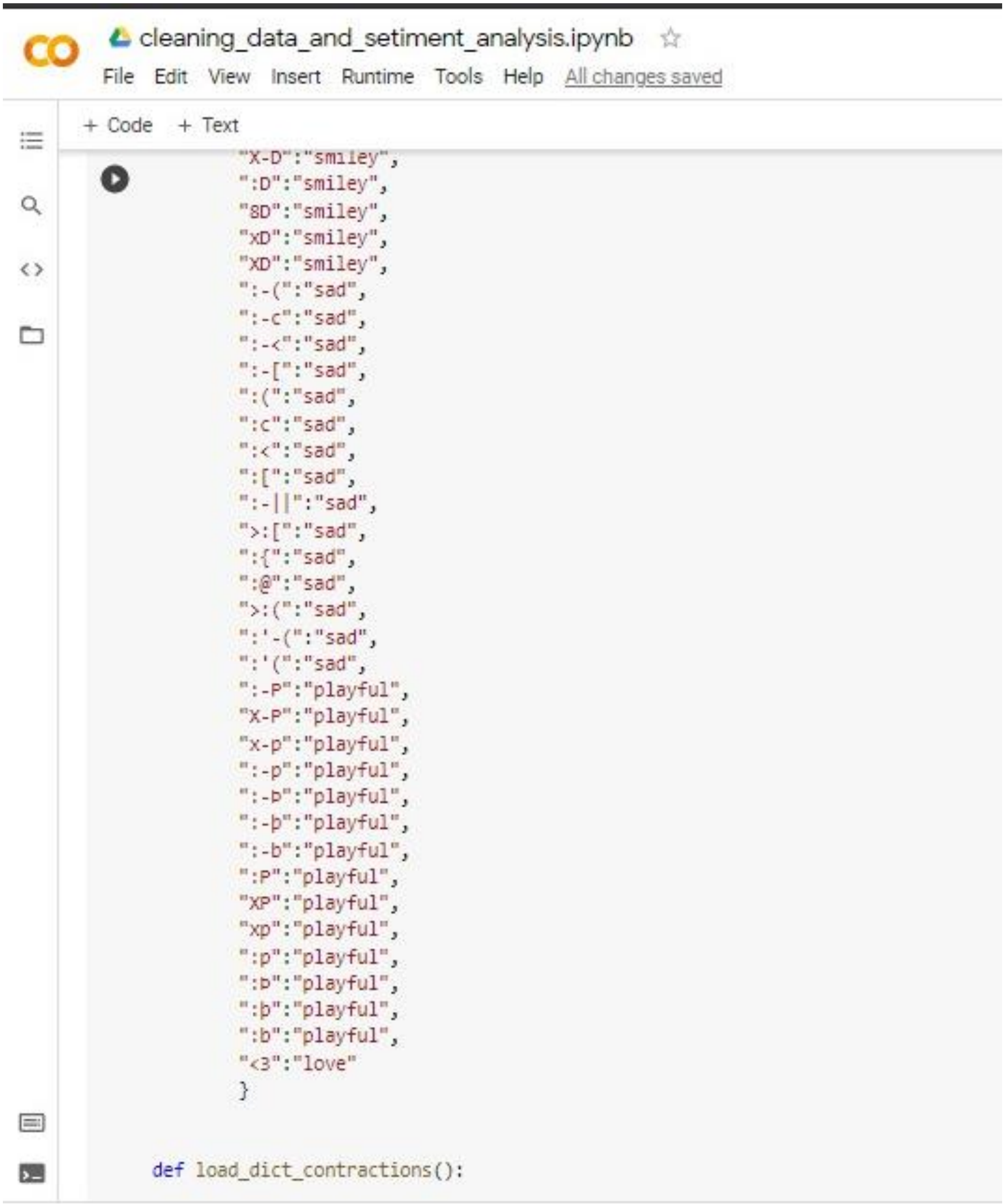
```
[7] df.shape
```

```
[8] df.info()
```

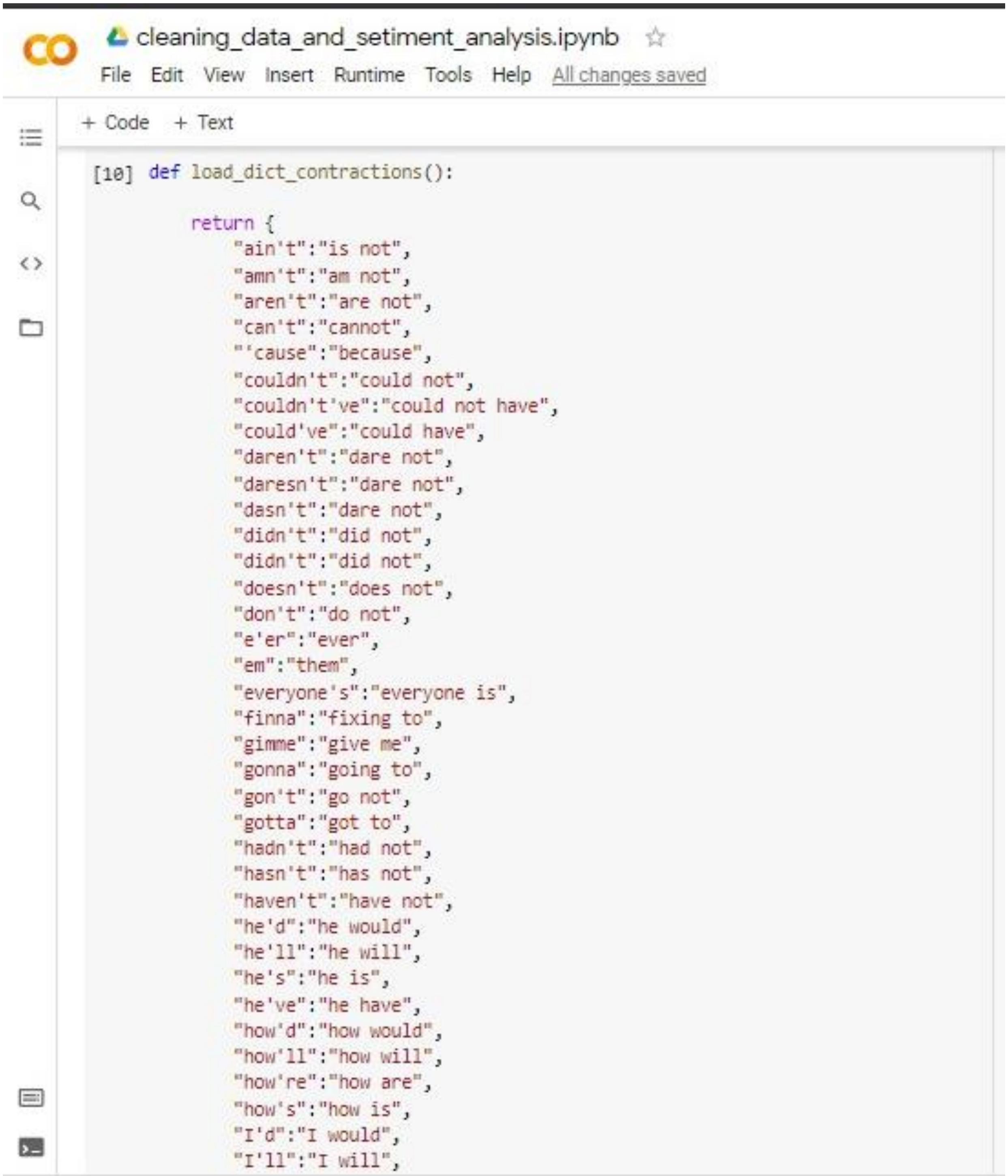
```
[9] !pip install emoji
```



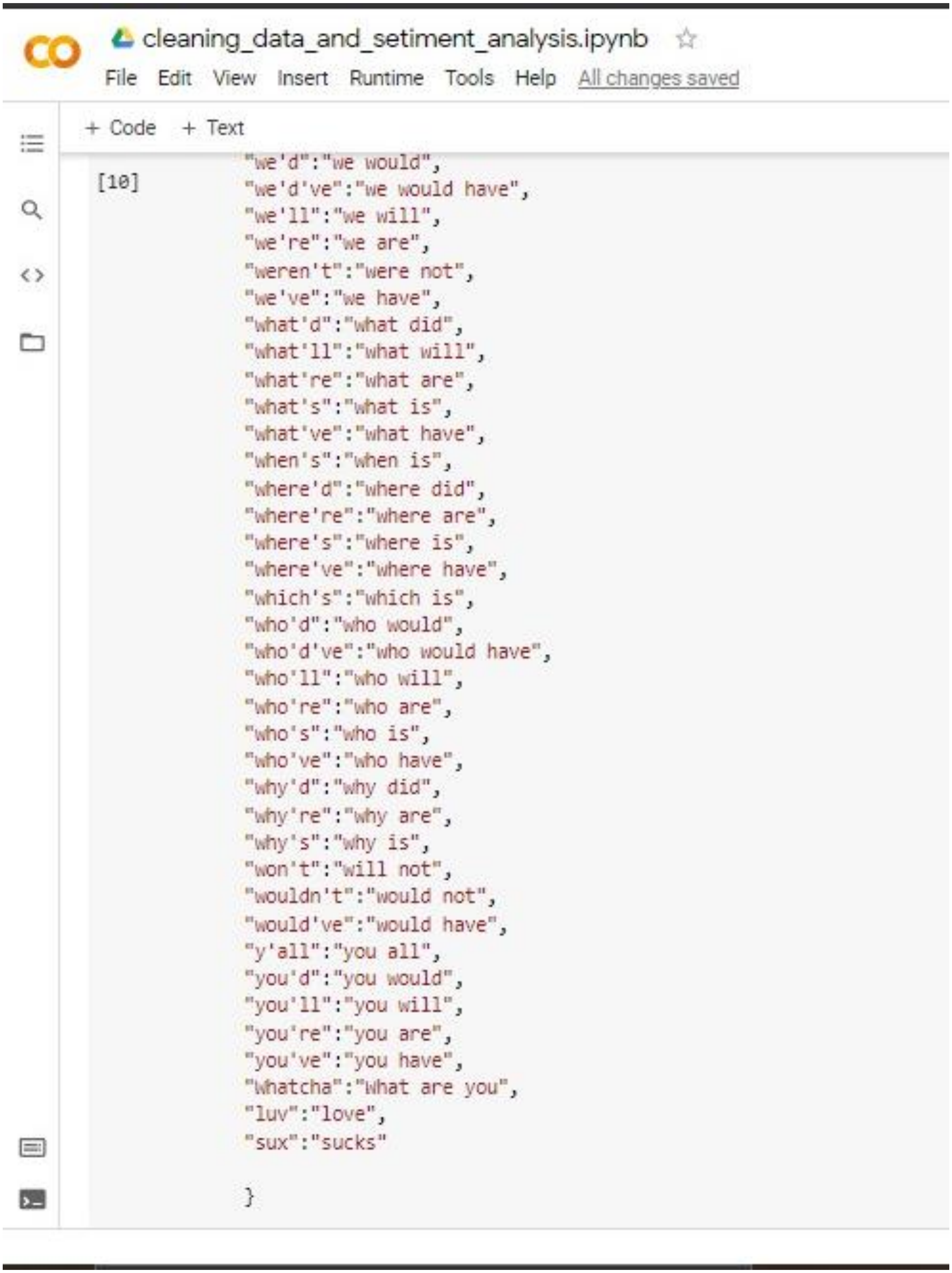
```
[10] from nltk.tokenize import word_tokenize
     from string import punctuation
     from nltk.corpus import stopwords

     import nltk
     import sys
     import os
     nltk.download('punkt')
     import csv
     import datetime
     from bs4 import BeautifulSoup
     import re
     import itertools
     import emoji
     def load_dict_smileys():

         return {
             ":-)":"smiley",
             ":-]":"smiley",
             ":-3":"smiley",
             ":->":"smiley",
             "8-)":"smiley",
             ":-}":"smiley",
             ":)":"smiley",
             ":]":"smiley",
             ":3":"smiley",
             ":>":"smiley",
             "8)":"smiley",
             ":}":"smiley",
             ":o)":"smiley",
             ":c)":"smiley",
             ":^)":"smiley",
             "=]":"smiley",
             "=)":"smiley",
             ":-))":"smiley",
             ":-D":"smiley",
             "8-D":"smiley",
             "x-D":"smiley",
```



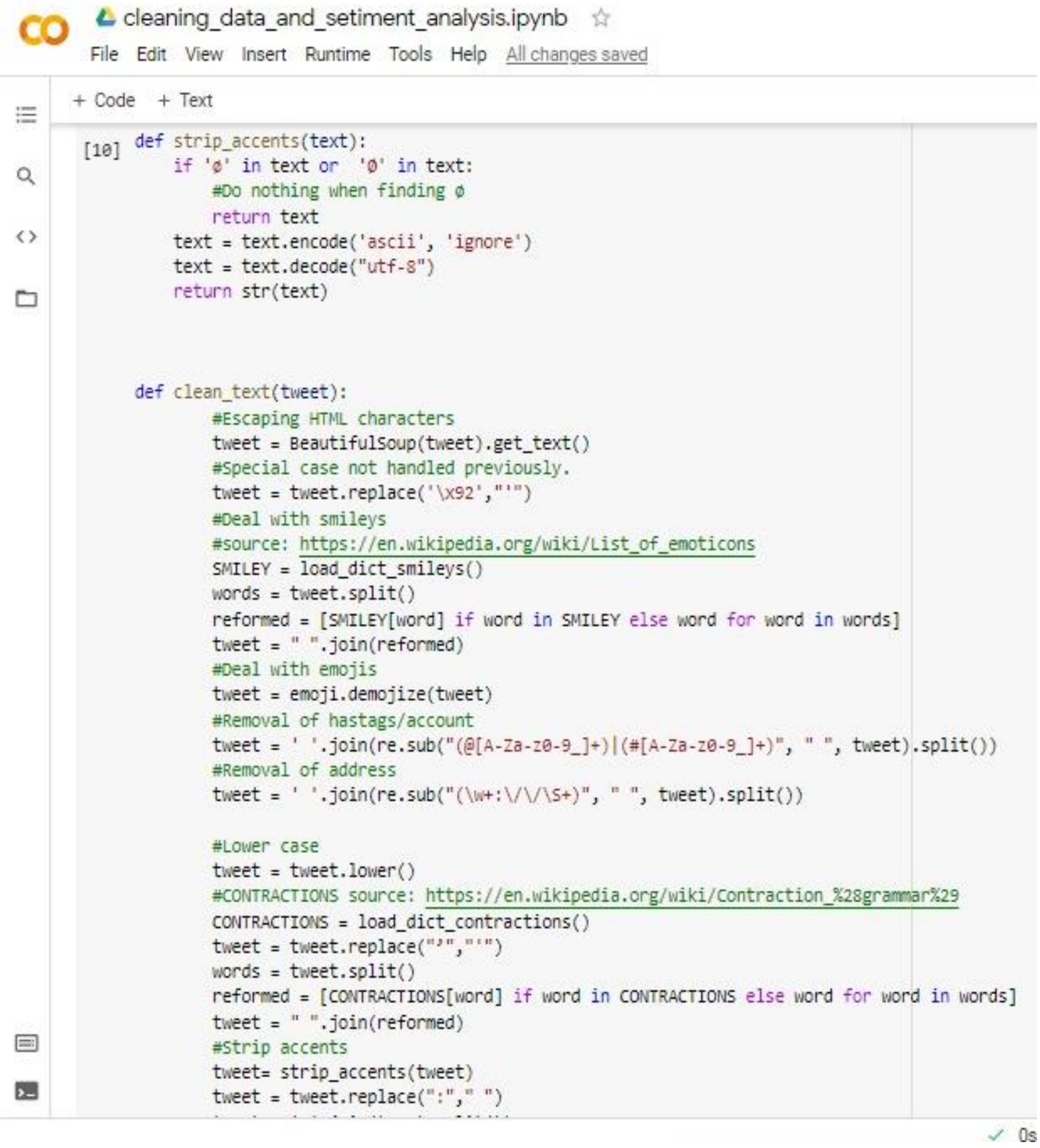
```
            "X-D":"smiley",
            ":D":"smiley",
            "8D":"smiley",
            "xD":"smiley",
            "XD":"smiley",
            ":-(":"sad",
            ":-c":"sad",
            ":-<":"sad",
            ":-[":"sad",
            ":(":"sad",
            ":c":"sad",
            ":<":"sad",
            ":[":"sad",
            ":-||":"sad",
            ">:[":"sad",
            ":{":"sad",
            ":@":"sad",
            ">:(":"sad",
            ":'-(":"sad",
            ":'(":"sad",
            ":-P":"playful",
            "X-P":"playful",
            "x-p":"playful",
            ":-p":"playful",
            ":-Þ":"playful",
            ":-þ":"playful",
            ":-b":"playful",
            ":P":"playful",
            "XP":"playful",
            "xp":"playful",
            ":p":"playful",
            ":Þ":"playful",
            ":þ":"playful",
            ":b":"playful",
            "<3":"love"
            }


    def load_dict_contractions():
```



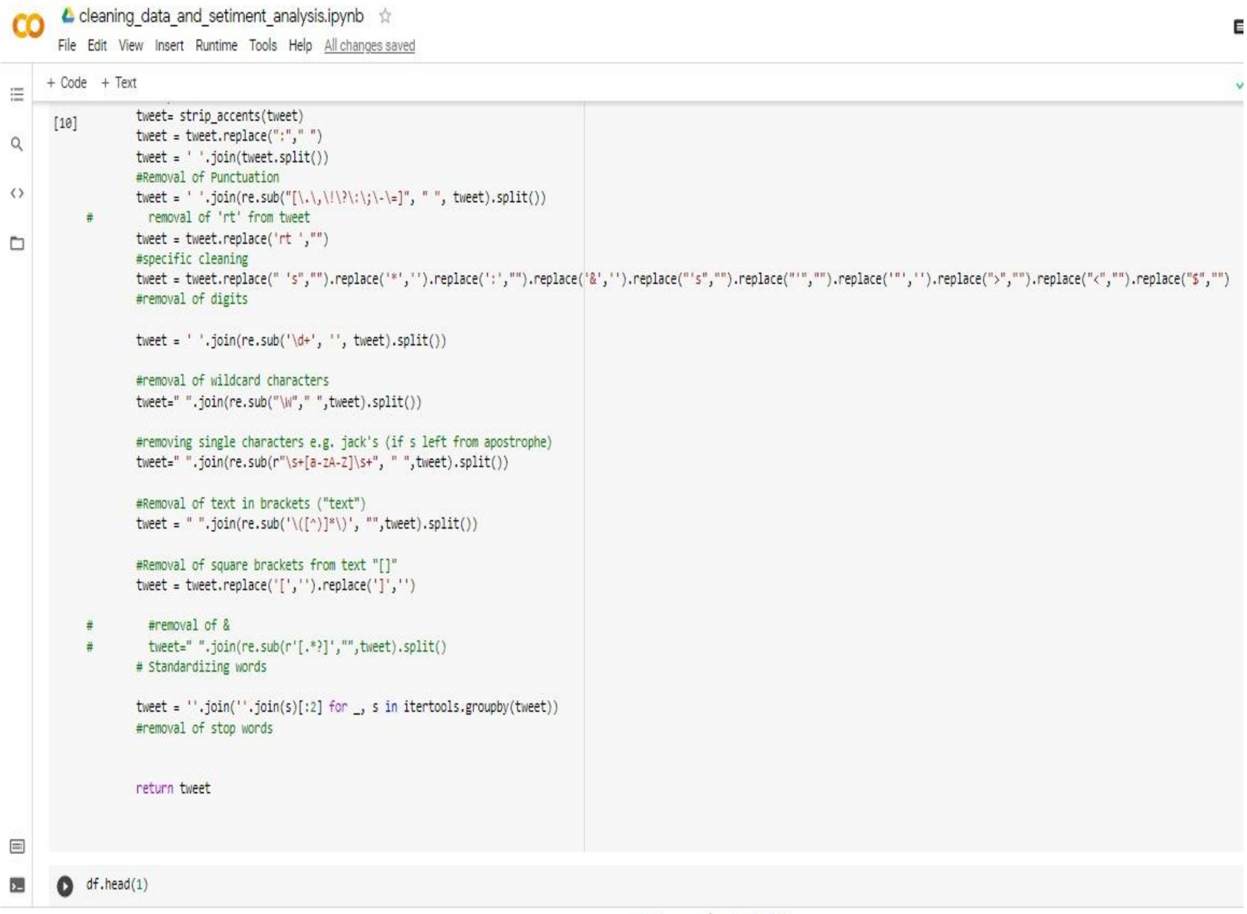
```
def load_dict_contractions():

    return {
        "ain't":"is not",
        "amn't":"am not",
        "aren't":"are not",
        "can't":"cannot",
        "'cause":"because",
        "couldn't":"could not",
        "couldn't've":"could not have",
        "could've":"could have",
        "daren't":"dare not",
        "daresn't":"dare not",
        "dasn't":"dare not",
        "didn't":"did not",
        "didn't":"did not",
        "doesn't":"does not",
        "don't":"do not",
        "e'er":"ever",
        "em":"them",
        "everyone's":"everyone is",
        "finna":"fixing to",
        "gimme":"give me",
        "gonna":"going to",
        "gon't":"go not",
        "gotta":"got to",
        "hadn't":"had not",
        "hasn't":"has not",
        "haven't":"have not",
        "he'd":"he would",
        "he'll":"he will",
        "he's":"he is",
        "he've":"he have",
        "how'd":"how would",
        "how'll":"how will",
        "how're":"how are",
        "how's":"how is",
        "I'd":"I would",
        "I'll":"I will",
```



```
"we'd":"we would",
"we'd've":"we would have",
"we'll":"we will",
"we're":"we are",
"weren't":"were not",
"we've":"we have",
"what'd":"what did",
"what'll":"what will",
"what're":"what are",
"what's":"what is",
"what've":"what have",
"when's":"when is",
"where'd":"where did",
"where're":"where are",
"where's":"where is",
"where've":"where have",
"which's":"which is",
"who'd":"who would",
"who'd've":"who would have",
"who'll":"who will",
"who're":"who are",
"who's":"who is",
"who've":"who have",
"why'd":"why did",
"why're":"why are",
"why's":"why is",
"won't":"will not",
"wouldn't":"would not",
"would've":"would have",
"y'all":"you all",
"you'd":"you would",
"you'll":"you will",
"you're":"you are",
"you've":"you have",
"Whatcha":"What are you",
"luv":"love",
"sux":"sucks"

}
```

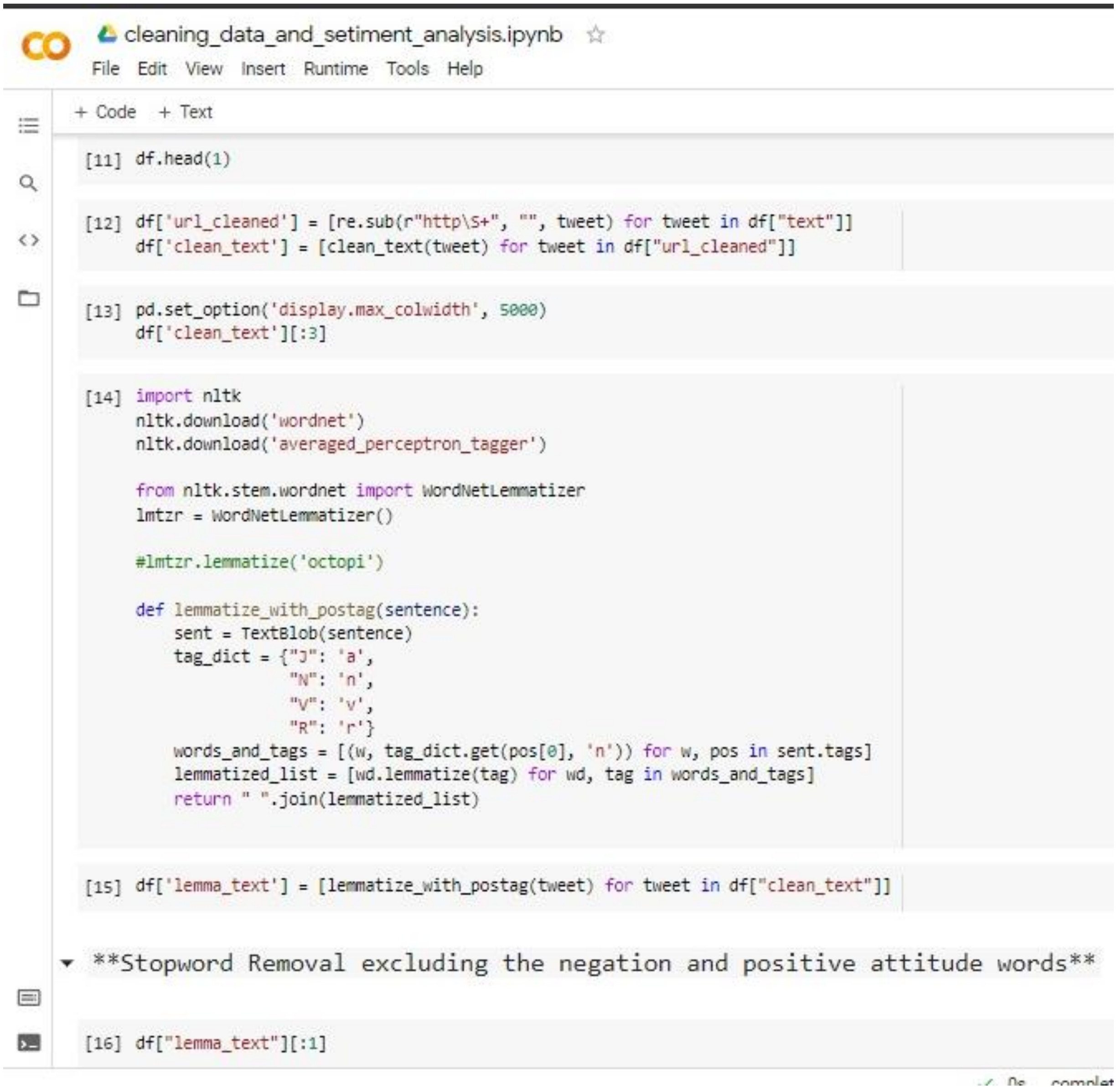

cleaning_data_and_setiment_analysis.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
[10]
✓ 0s

```
def strip_accents(text):
    if 'ø' in text or  'Ø' in text:
        #Do nothing when finding ø
        return text
    text = text.encode('ascii', 'ignore')
    text = text.decode("utf-8")
    return str(text)



def clean_text(tweet):
        #Escaping HTML characters
        tweet = BeautifulSoup(tweet).get_text()
        #Special case not handled previously.
        tweet = tweet.replace('\x92',"'")
        #Deal with smileys
        #source: https://en.wikipedia.org/wiki/List_of_emoticons
        SMILEY = load_dict_smileys()
        words = tweet.split()
        reformed = [SMILEY[word] if word in SMILEY else word for word in words]
        tweet = " ".join(reformed)
        #Deal with emojis
        tweet = emoji.demojize(tweet)
        #Removal of hastags/account
        tweet = ' '.join(re.sub("(@[A-Za-z0-9_]+)|(#[A-Za-z0-9_]+)", " ", tweet).split())
        #Removal of address
        tweet = ' '.join(re.sub("(\w+:\/\/\S+)", " ", tweet).split())

        #Lower case
        tweet = tweet.lower()
        #CONTRACTIONS source: https://en.wikipedia.org/wiki/Contraction_%28grammar%29
        CONTRACTIONS = load_dict_contractions()
        tweet = tweet.replace("’","'")
        words = tweet.split()
        reformed = [CONTRACTIONS[word] if word in CONTRACTIONS else word for word in words]
        tweet = " ".join(reformed)
        #Strip accents
        tweet= strip_accents(tweet)
        tweet = tweet.replace(":"," ")
```


cleaning_data_and_setiment_analysis.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text

```
[10]        tweet= strip_accents(tweet)
            tweet = tweet.replace(":"," ")
            tweet = ' '.join(tweet.split())
            #Removal of Punctuation
            tweet = ' '.join(re.sub("[\.\,\!\?\:\;\-\=]", " ", tweet).split())
    #         removal of 'rt' from tweet
            tweet = tweet.replace('rt ',"")
            #specific cleaning
            tweet = tweet.replace(" 's","").replace('*','').replace(':',"").replace('&','').replace("'s","").replace("'","").replace('"','').replace(">","").replace("<","").replace("$","")
            #removal of digits

            tweet = ' '.join(re.sub('\d+', '', tweet).split())

            #removal of wildcard characters
            tweet=" ".join(re.sub("\W"," ",tweet).split())

            #removing single characters e.g. jack's (if s left from apostrophe)
            tweet=" ".join(re.sub(r"\s+[a-zA-Z]\s+", " ",tweet).split())

            #Removal of text in brackets ("text")
            tweet = " ".join(re.sub('\([^)]*\)', "",tweet).split())

            #Removal of square brackets from text "[]"
            tweet = tweet.replace('[','').replace(']','')

    #         #removal of &
    #         tweet=" ".join(re.sub(r'[.*?]',"",tweet).split()
            # Standardizing words

            tweet = ''.join(''.join(s)[:2] for _, s in itertools.groupby(tweet))
            #removal of stop words


            return tweet
```

```
df.head(1)
```


CO
cleaning_data_and_setiment_analysis.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
✓ 0s complet

```
[11] df.head(1)
```

```
[12] df['url_cleaned'] = [re.sub(r"http\S+", "", tweet) for tweet in df["text"]]
     df['clean_text'] = [clean_text(tweet) for tweet in df["url_cleaned"]]
```

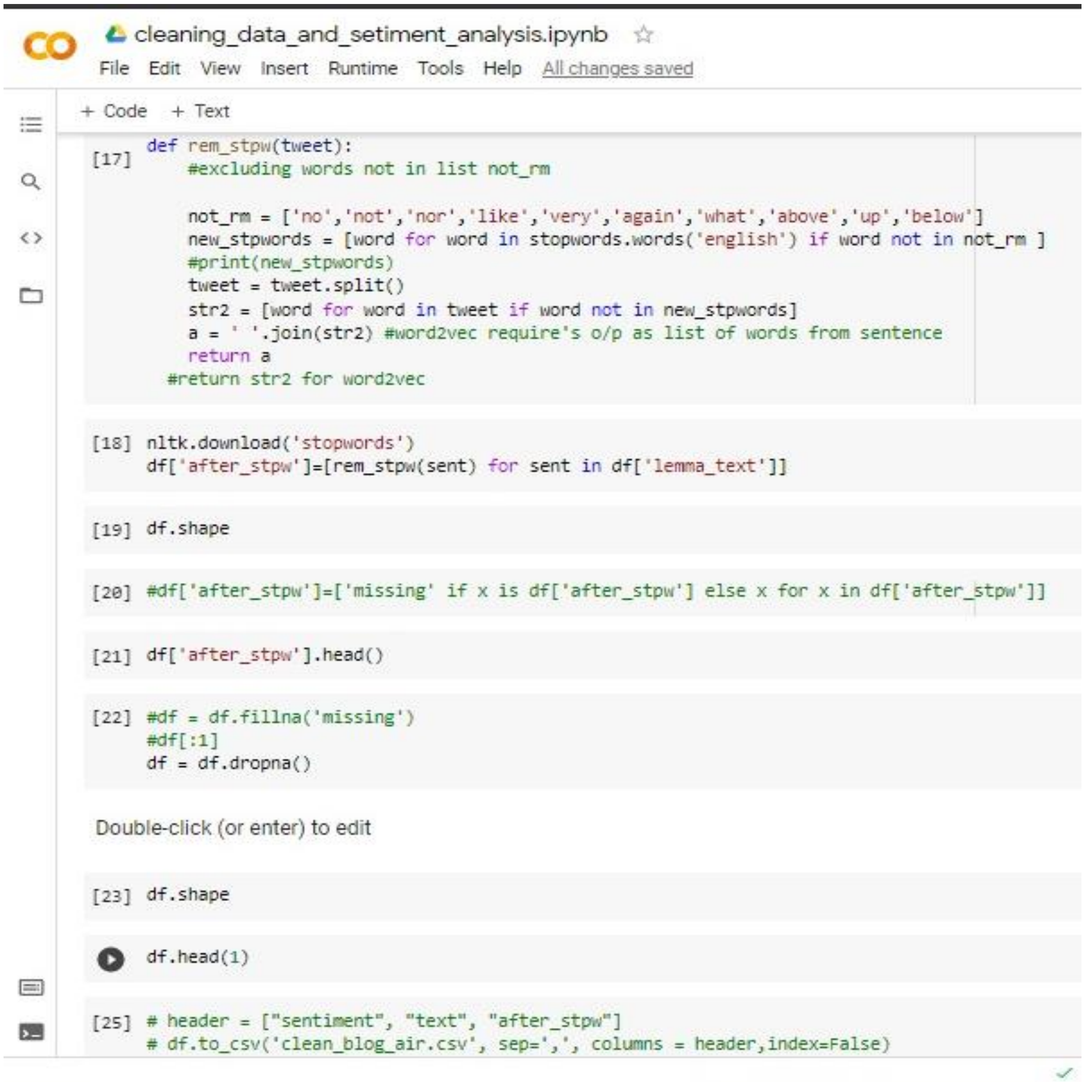
```
[13] pd.set_option('display.max_colwidth', 5000)
     df['clean_text'][:3]
```

```
[14] import nltk
     nltk.download('wordnet')
     nltk.download('averaged_perceptron_tagger')

     from nltk.stem.wordnet import WordNetLemmatizer
     lmtzr = WordNetLemmatizer()

     #lmtzr.lemmatize('octopi')

     def lemmatize_with_postag(sentence):
         sent = TextBlob(sentence)
         tag_dict = {"J": 'a',
                     "N": 'n',
                     "V": 'v',
                     "R": 'r'}
         words_and_tags = [(w, tag_dict.get(pos[0], 'n')) for w, pos in sent.tags]
         lemmatized_list = [wd.lemmatize(tag) for wd, tag in words_and_tags]
         return " ".join(lemmatized_list)
```

```
[15] df['lemma_text'] = [lemmatize_with_postag(tweet) for tweet in df["clean_text"]]
```

Stopword Removal excluding the negation and positive attitude words

```
[16] df["lemma_text"][:1]
```


cleaning_data_and_setiment_analysis.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text

```
[17] def rem_stpw(tweet):
        #excluding words not in list not_rm

        not_rm = ['no','not','nor','like','very','again','what','above','up','below']
        new_stpwords = [word for word in stopwords.words('english') if word not in not_rm ]
        #print(new_stpwords)
        tweet = tweet.split()
        str2 = [word for word in tweet if word not in new_stpwords]
        a = ' '.join(str2) #word2vec require's o/p as list of words from sentence
        return a
      #return str2 for word2vec
```

```
[18] nltk.download('stopwords')
     df['after_stpw']=[rem_stpw(sent) for sent in df['lemma_text']]
```

```
[19] df.shape
```

```
[20] #df['after_stpw']=['missing' if x is df['after_stpw'] else x for x in df['after_stpw']]
```

```
[21] df['after_stpw'].head()
```

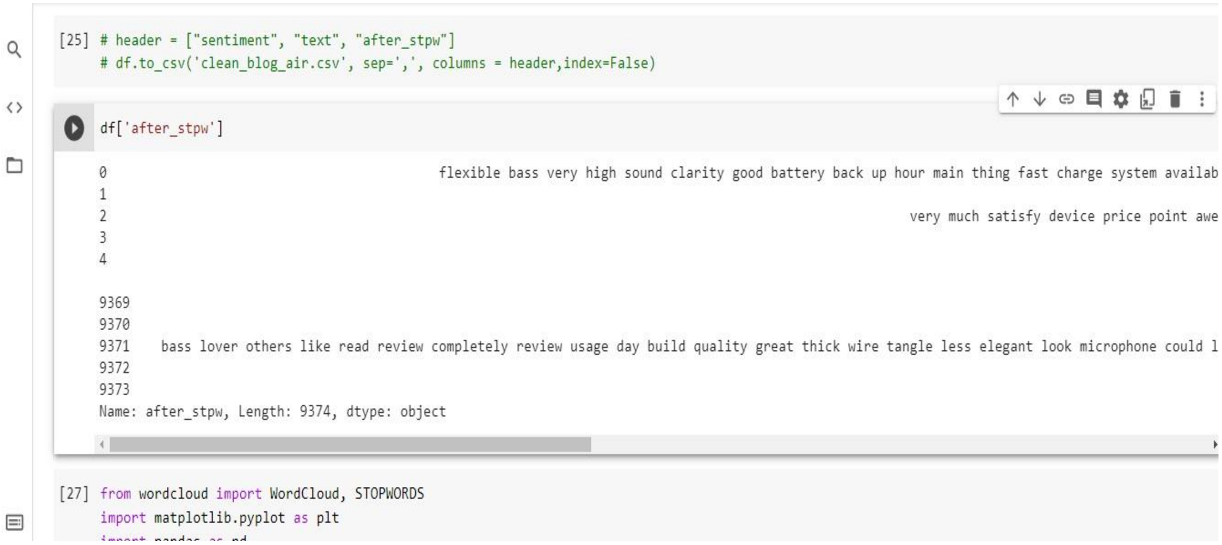
```
[22] #df = df.fillna('missing')
     #df[:1]
     df = df.dropna()
```

Double-click (or enter) to edit

```
[23] df.shape
```

```
df.head(1)
```

```
[25] # header = ["sentiment", "text", "after_stpw"]
     # df.to_csv('clean_blog_air.csv', sep=',', columns = header,index=False)
```



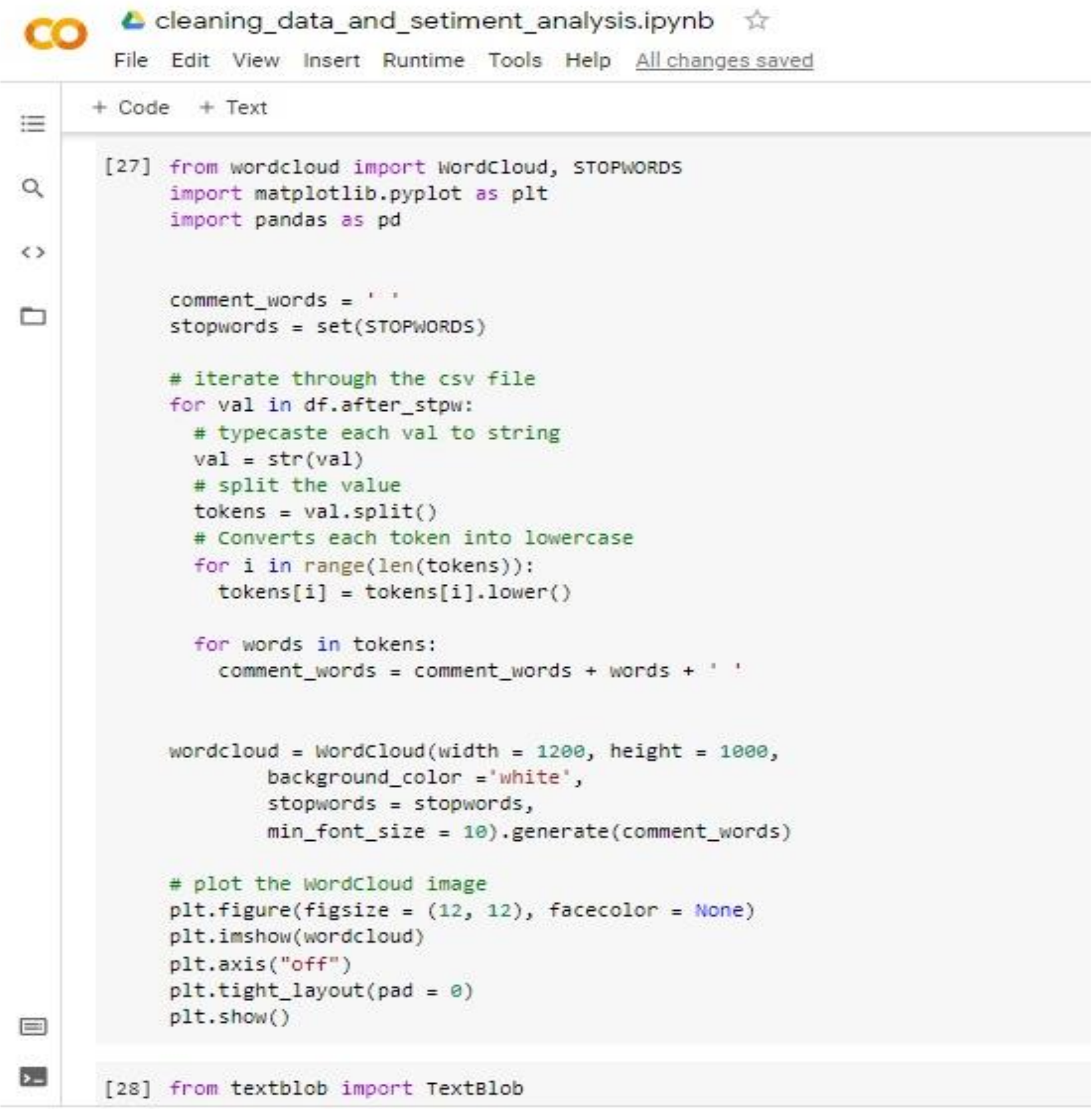
```
[25] # header = ["sentiment", "text", "after_stpw"]
     # df.to_csv('clean_blog_air.csv', sep=',', columns = header,index=False)
```

```
df['after_stpw']
```

```
0                                            flexible bass very high sound clarity good battery back up hour main thing fast charge system availab
1
2                                                                                                                    very much satisfy device price point awe
3
4

9369
9370
9371    bass lover others like read review completely review usage day build quality great thick wire tangle less elegant look microphone could l
9372
9373
Name: after_stpw, Length: 9374, dtype: object
```

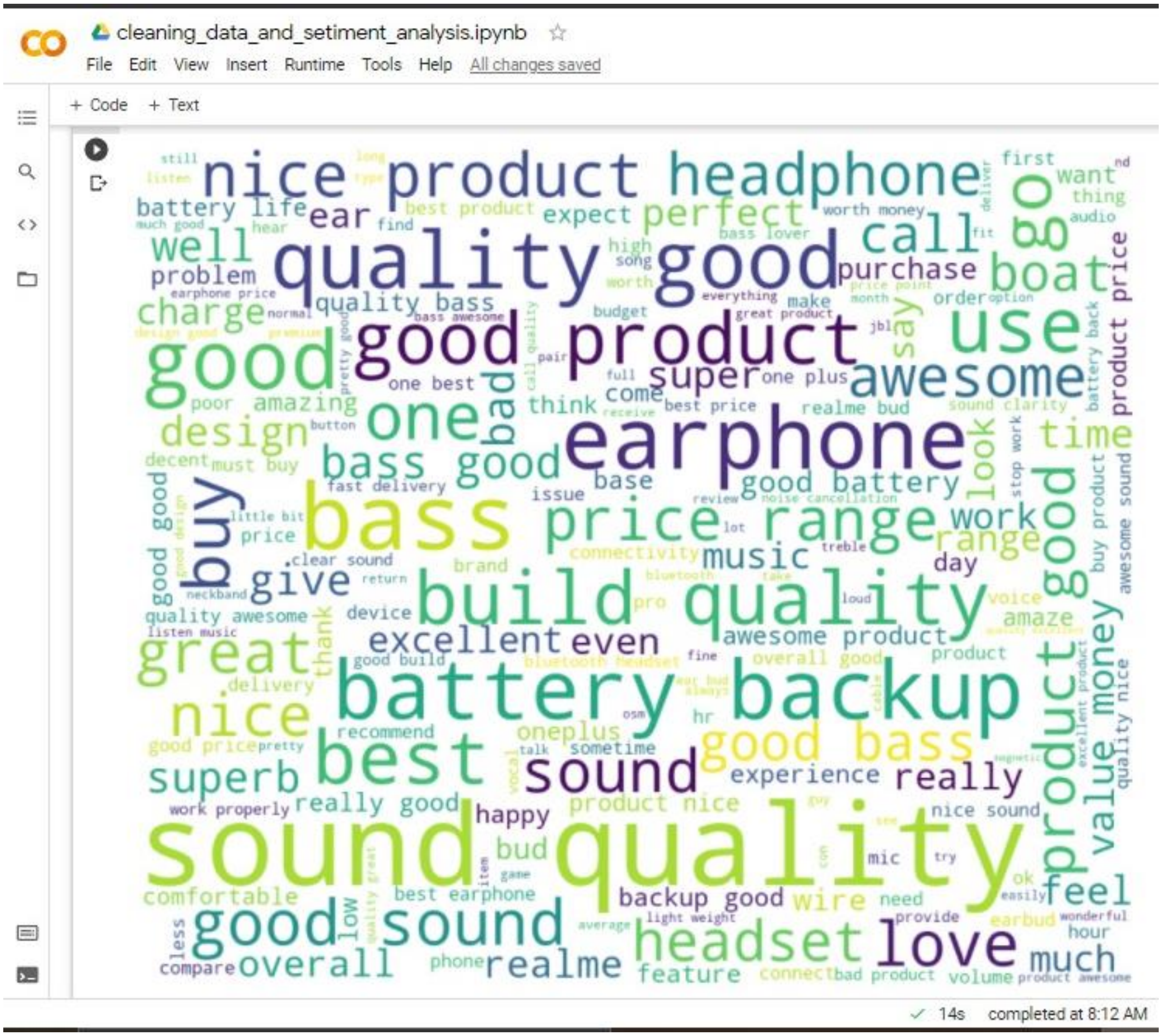
```
[27] from wordcloud import WordCloud, STOPWORDS
     import matplotlib.pyplot as plt
     import pandas as pd
```

Word cloud of the reviews:

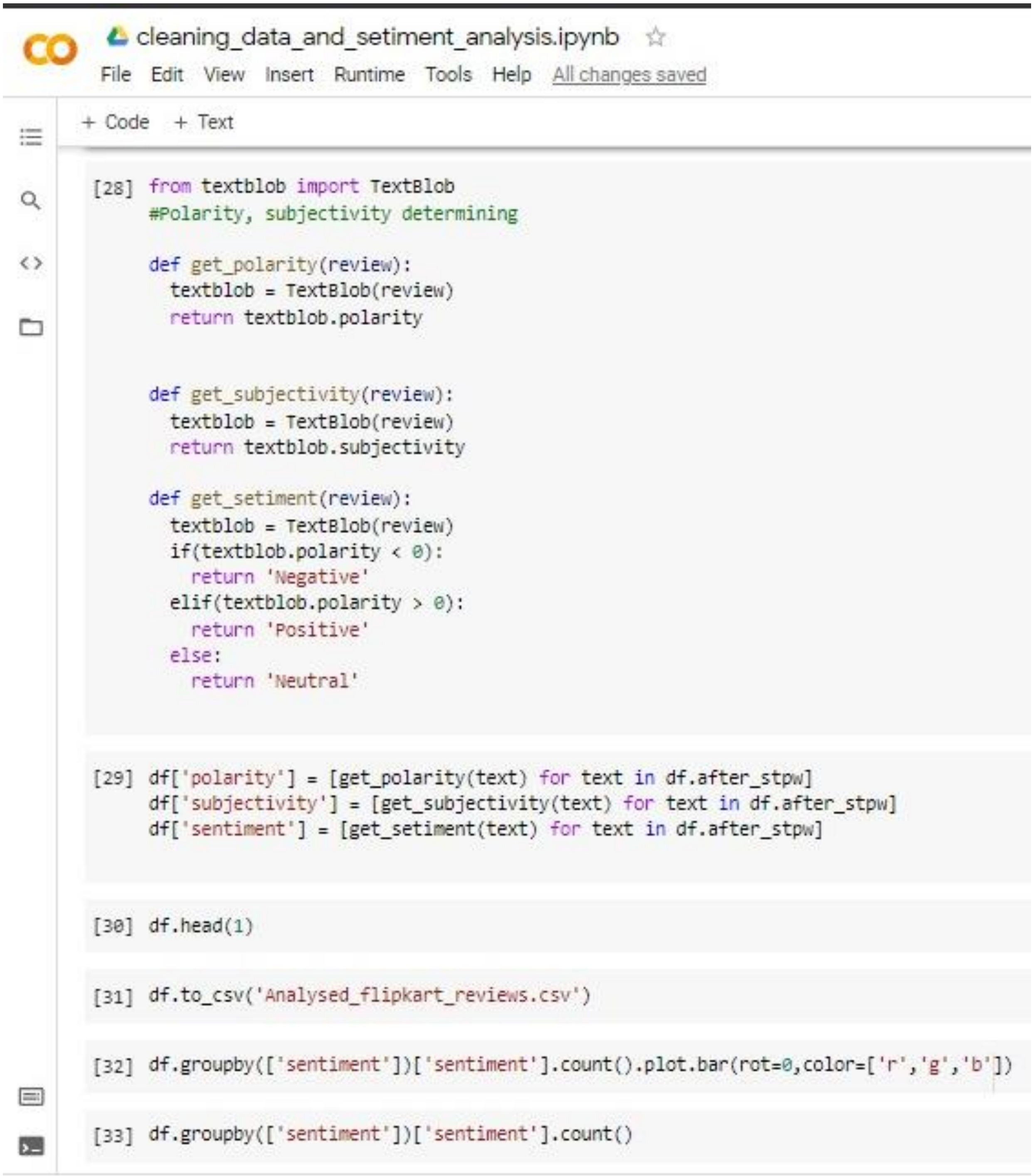


```
[27] from wordcloud import WordCloud, STOPWORDS
     import matplotlib.pyplot as plt
     import pandas as pd


     comment_words = ' '
     stopwords = set(STOPWORDS)

     # iterate through the csv file
     for val in df.after_stpw:
       # typecaste each val to string
       val = str(val)
       # split the value
       tokens = val.split()
       # Converts each token into lowercase
       for i in range(len(tokens)):
         tokens[i] = tokens[i].lower()

       for words in tokens:
         comment_words = comment_words + words + ' '


     wordcloud = WordCloud(width = 1200, height = 1000,
             background_color ='white',
             stopwords = stopwords,
             min_font_size = 10).generate(comment_words)

     # plot the WordCloud image
     plt.figure(figsize = (12, 12), facecolor = None)
     plt.imshow(wordcloud)
     plt.axis("off")
     plt.tight_layout(pad = 0)
     plt.show()
```

```
[28] from textblob import TextBlob
```


cleaning_data_and_setiment_analysis.ipynb
File
Edit
View
Insert
Runtime
Tools
Help
All changes saved
+ Code
+ Text
nice product
headphone
quality
good
call
boat
use
good product
awesome
earphone
bass
price range
build quality
battery backup
sound quality
sound
best
superb
great
nice
buy
one
design
good sound
headset
love
realme
overall
product good
value money
go
perfect
well
charge
problem
battery life
ear
expect
purchase
super
bass good
good battery
work
range
music
give
excellent even
awesome product
good bass
experience
really
really good
happy
bud
backup good
wire
feel
much
feature
comfortable
best earphone
oneplus
recommend
delivery
time
look
say
product price
buy product
awesome sound
quality nice
✓ 14s
completed at 8:12 AM

Sentiment analysis using TextBlob to find polarity and subjectivity:


cleaning_data_and_setiment_analysis.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text

```
[28] from textblob import TextBlob
     #Polarity, subjectivity determining

     def get_polarity(review):
       textblob = TextBlob(review)
       return textblob.polarity


     def get_subjectivity(review):
       textblob = TextBlob(review)
       return textblob.subjectivity

     def get_setiment(review):
       textblob = TextBlob(review)
       if(textblob.polarity < 0):
         return 'Negative'
       elif(textblob.polarity > 0):
         return 'Positive'
       else:
         return 'Neutral'
```

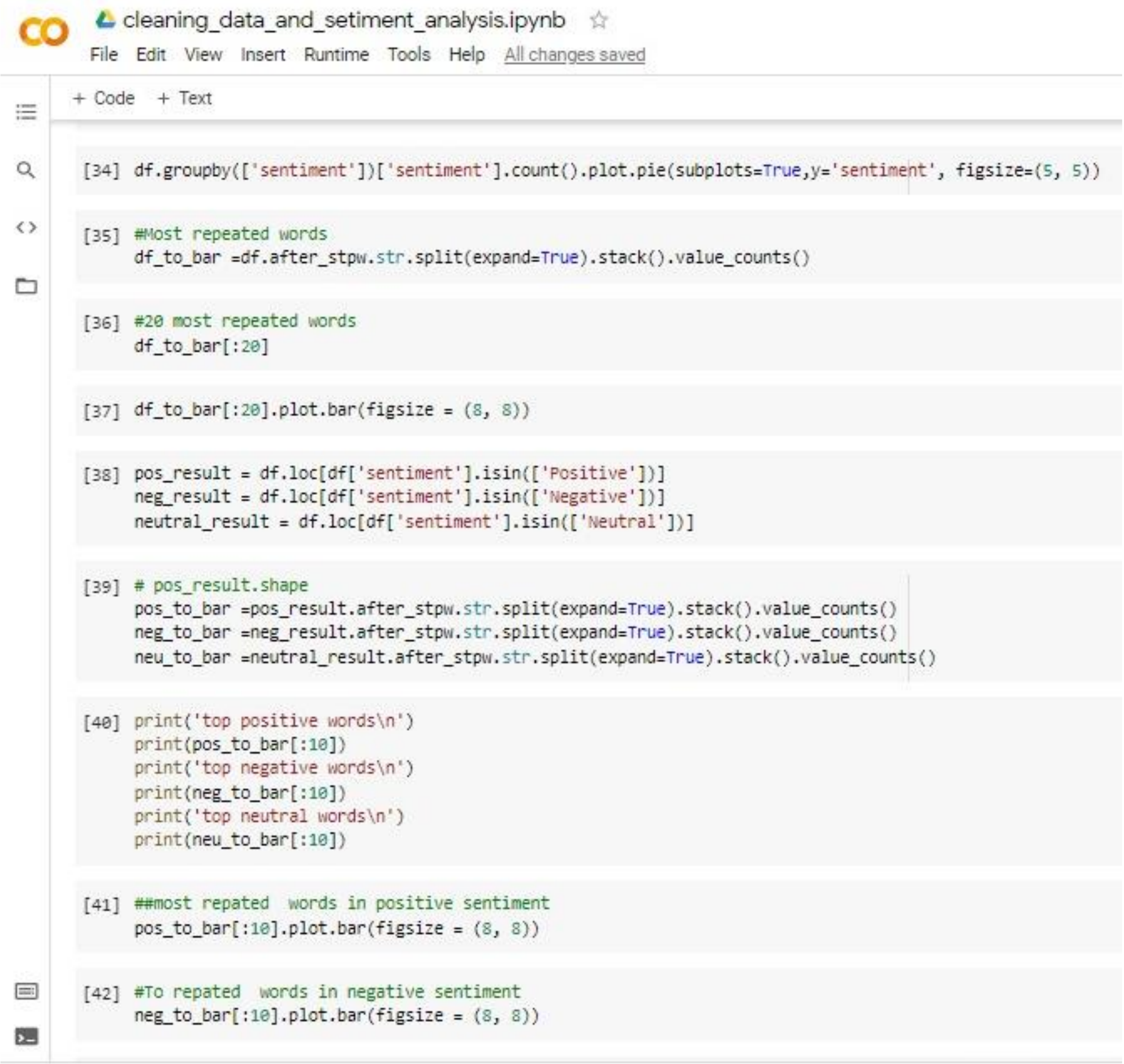
```
[29] df['polarity'] = [get_polarity(text) for text in df.after_stpw]
     df['subjectivity'] = [get_subjectivity(text) for text in df.after_stpw]
     df['sentiment'] = [get_setiment(text) for text in df.after_stpw]
```

```
[30] df.head(1)
```

```
[31] df.to_csv('Analysed_flipkart_reviews.csv')
```

```
[32] df.groupby(['sentiment'])['sentiment'].count().plot.bar(rot=0,color=['r','g','b'])
```

```
[33] df.groupby(['sentiment'])['sentiment'].count()
```



```
[34] df.groupby(['sentiment'])['sentiment'].count().plot.pie(subplots=True,y='sentiment', figsize=(5, 5))
```

```
[35] #Most repeated words
     df_to_bar =df.after_stpw.str.split(expand=True).stack().value_counts()
```

```
[36] #20 most repeated words
     df_to_bar[:20]
```

```
[37] df_to_bar[:20].plot.bar(figsize = (8, 8))
```

```
[38] pos_result = df.loc[df['sentiment'].isin(['Positive'])]
     neg_result = df.loc[df['sentiment'].isin(['Negative'])]
     neutral_result = df.loc[df['sentiment'].isin(['Neutral'])]
```

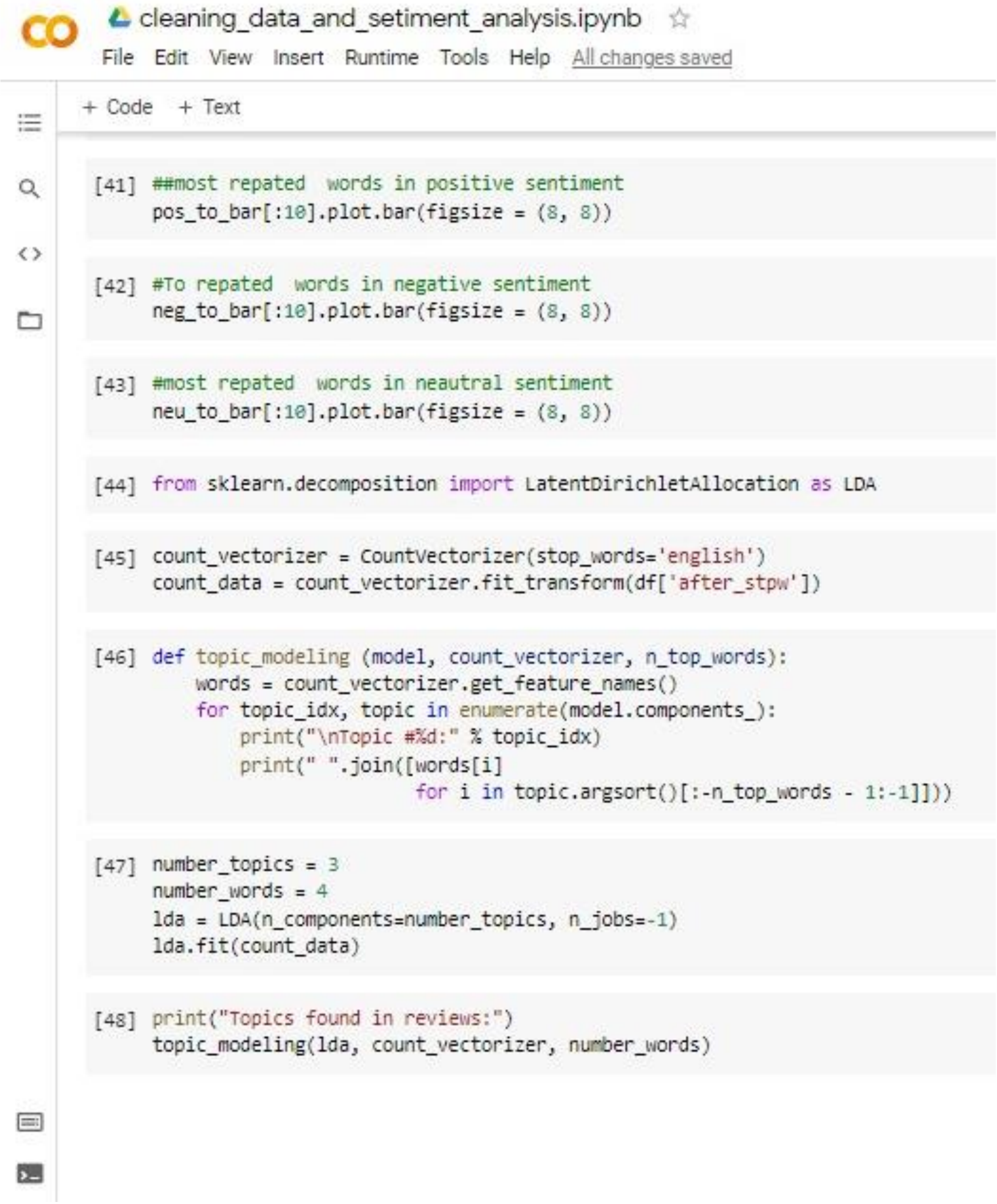
```
[39] # pos_result.shape
     pos_to_bar =pos_result.after_stpw.str.split(expand=True).stack().value_counts()
     neg_to_bar =neg_result.after_stpw.str.split(expand=True).stack().value_counts()
     neu_to_bar =neutral_result.after_stpw.str.split(expand=True).stack().value_counts()
```

```
[40] print('top positive words\n')
     print(pos_to_bar[:10])
     print('top negative words\n')
     print(neg_to_bar[:10])
     print('top neutral words\n')
     print(neu_to_bar[:10])
```

```
[41] ##most repated  words in positive sentiment
     pos_to_bar[:10].plot.bar(figsize = (8, 8))
```

```
[42] #To repated  words in negative sentiment
     neg_to_bar[:10].plot.bar(figsize = (8, 8))
```

Topic Modelling:


cleaning_data_and_setiment_analysis.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text

```
[41] ##most repated  words in positive sentiment
     pos_to_bar[:10].plot.bar(figsize = (8, 8))
```

```
[42] #To repated  words in negative sentiment
     neg_to_bar[:10].plot.bar(figsize = (8, 8))
```

```
[43] #most repated  words in neautral sentiment
     neu_to_bar[:10].plot.bar(figsize = (8, 8))
```

```
[44] from sklearn.decomposition import LatentDirichletAllocation as LDA
```

```
[45] count_vectorizer = CountVectorizer(stop_words='english')
     count_data = count_vectorizer.fit_transform(df['after_stpw'])
```

```
[46] def topic_modeling (model, count_vectorizer, n_top_words):
         words = count_vectorizer.get_feature_names()
         for topic_idx, topic in enumerate(model.components_):
             print("\nTopic #%d:" % topic_idx)
             print(" ".join([words[i]
                             for i in topic.argsort()[:-n_top_words - 1:-1]]))
```

```
[47] number_topics = 3
     number_words = 4
     lda = LDA(n_components=number_topics, n_jobs=-1)
     lda.fit(count_data)
```

```
[48] print("Topics found in reviews:")
     topic_modeling(lda, count_vectorizer, number_words)
```

APPENDIX

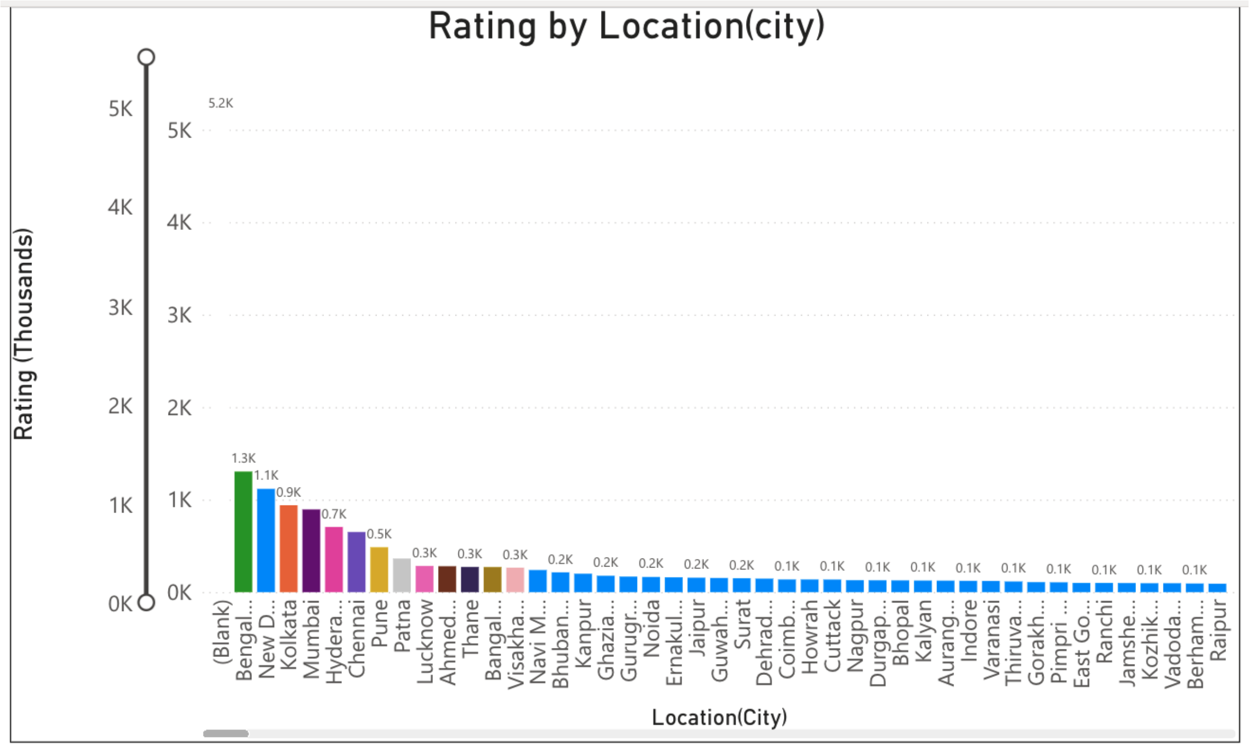


Figure 1: Count of Rating by location in India

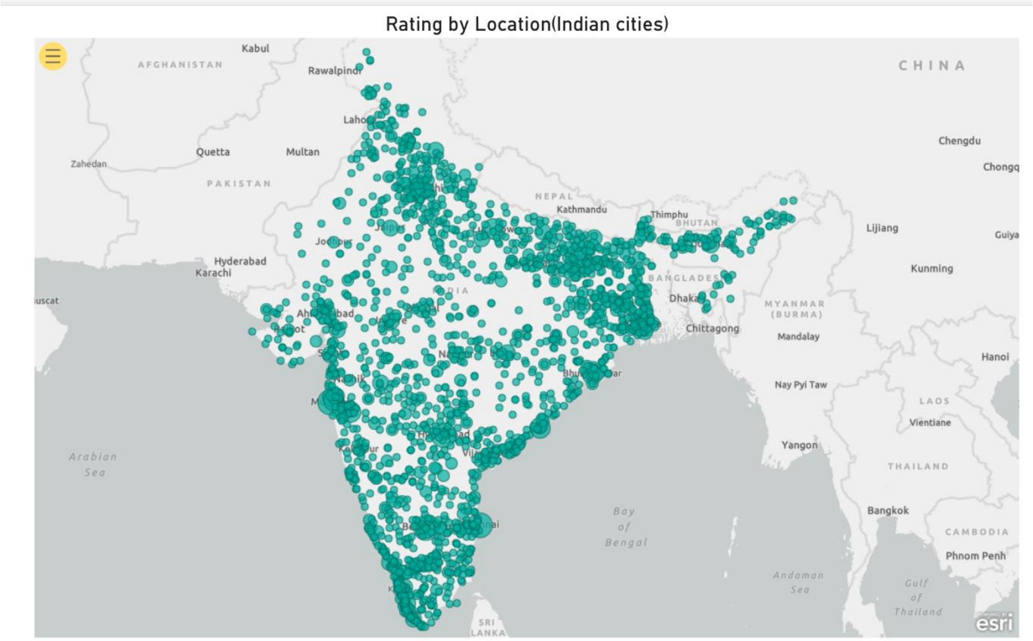


Figure 1.1: Indian cities map

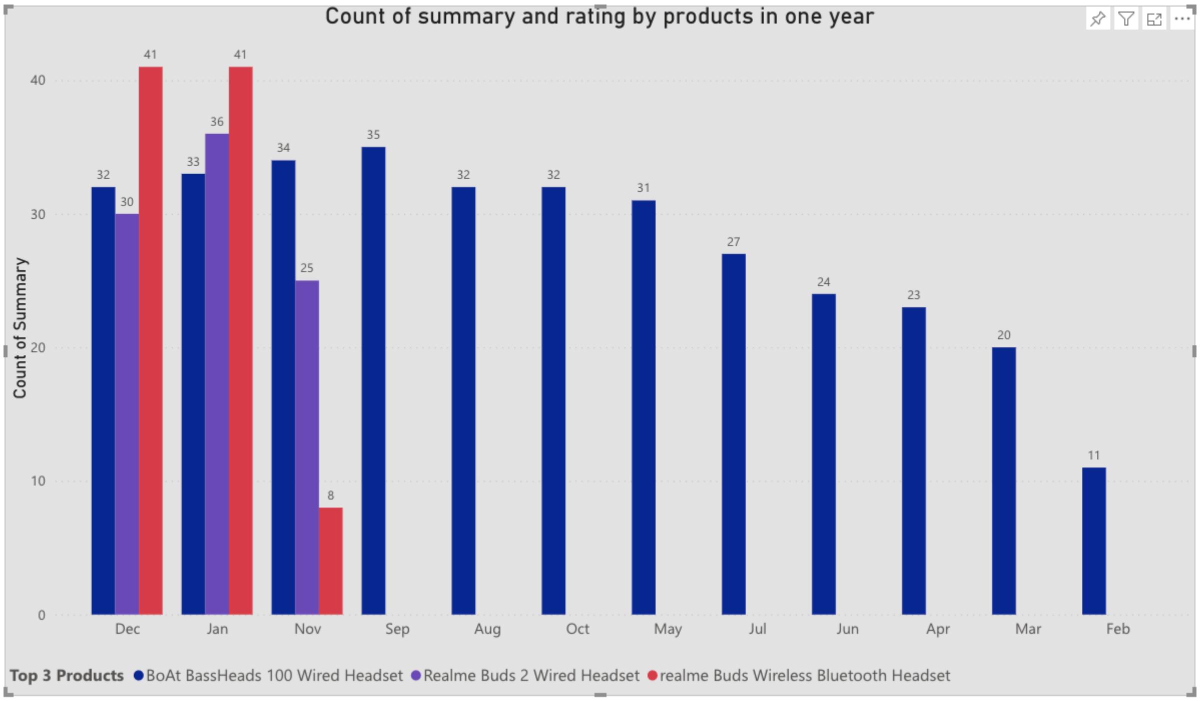


Figure 1.2: Count of summary and rating by products in one year

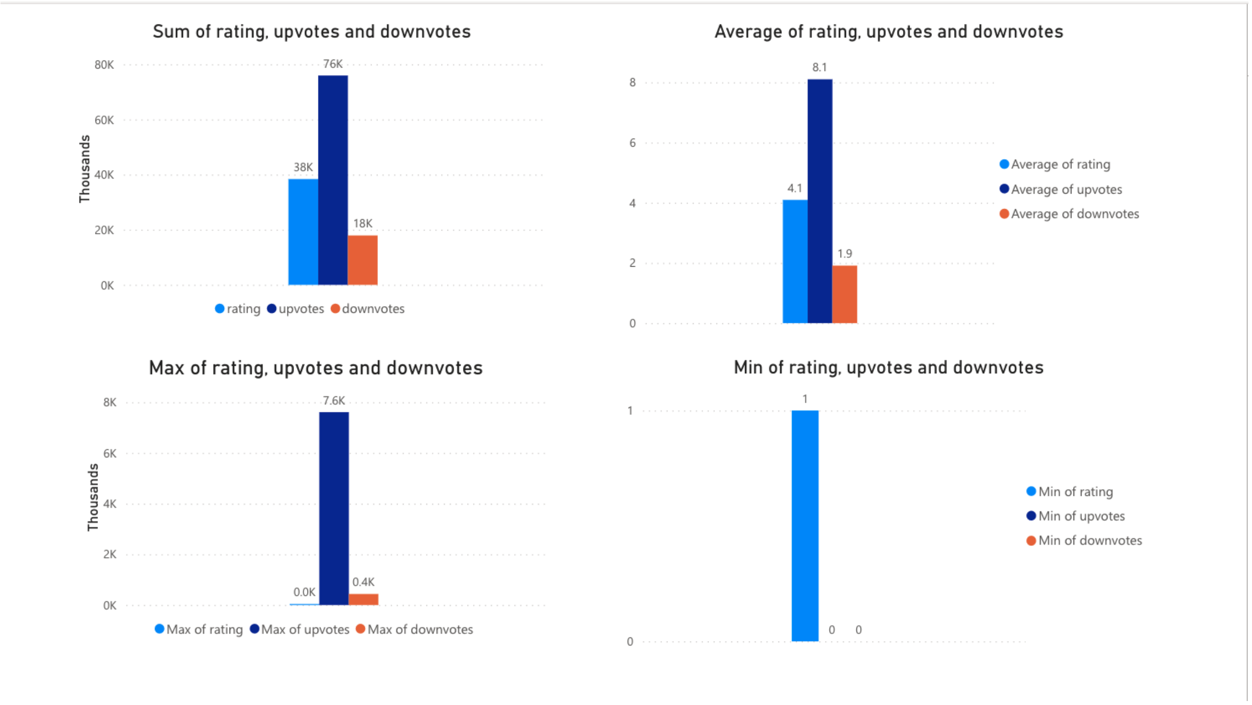


Figure 1.3: Sum, Average, Max and Min of rating, upvotes and downvotes

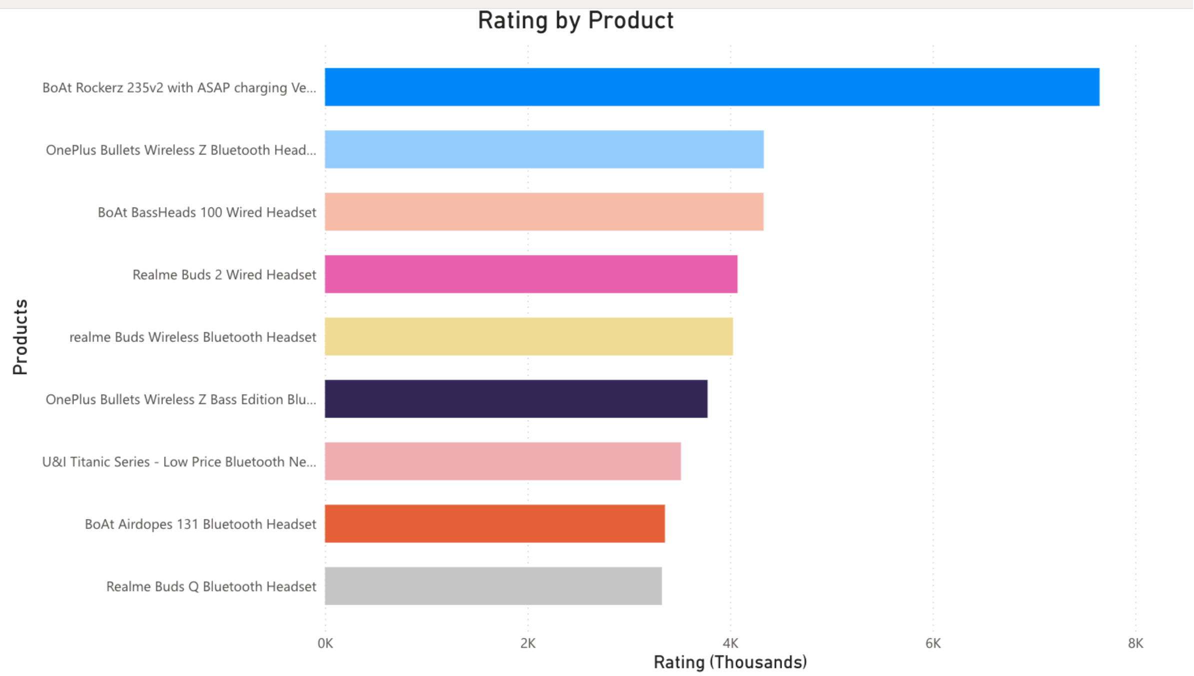


Figure 2: Ratings by product

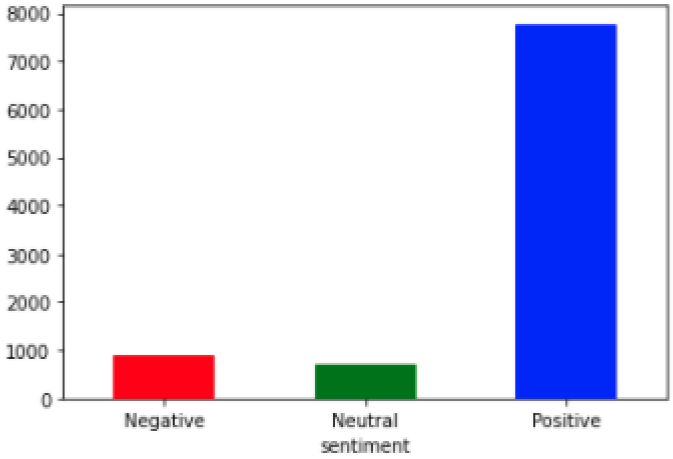


Figure 3: Review rate of product

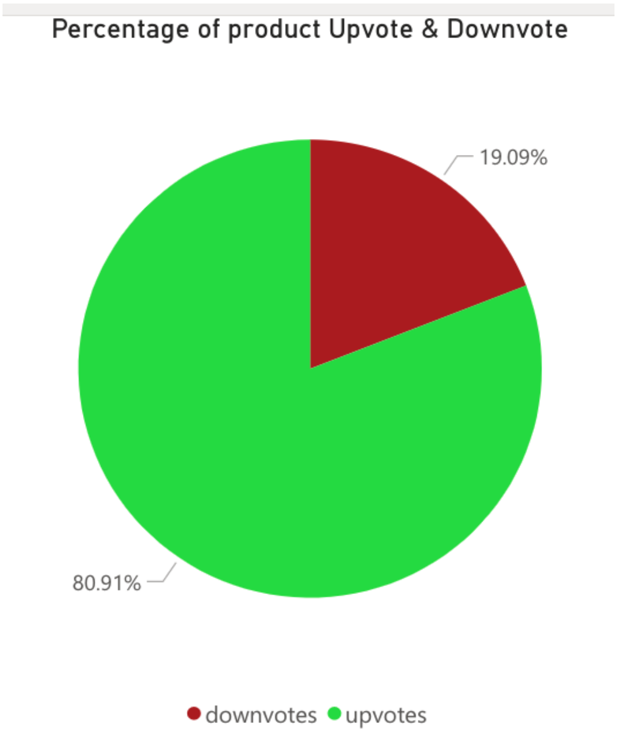


Figure 4: Percentage of product upvote and downvote

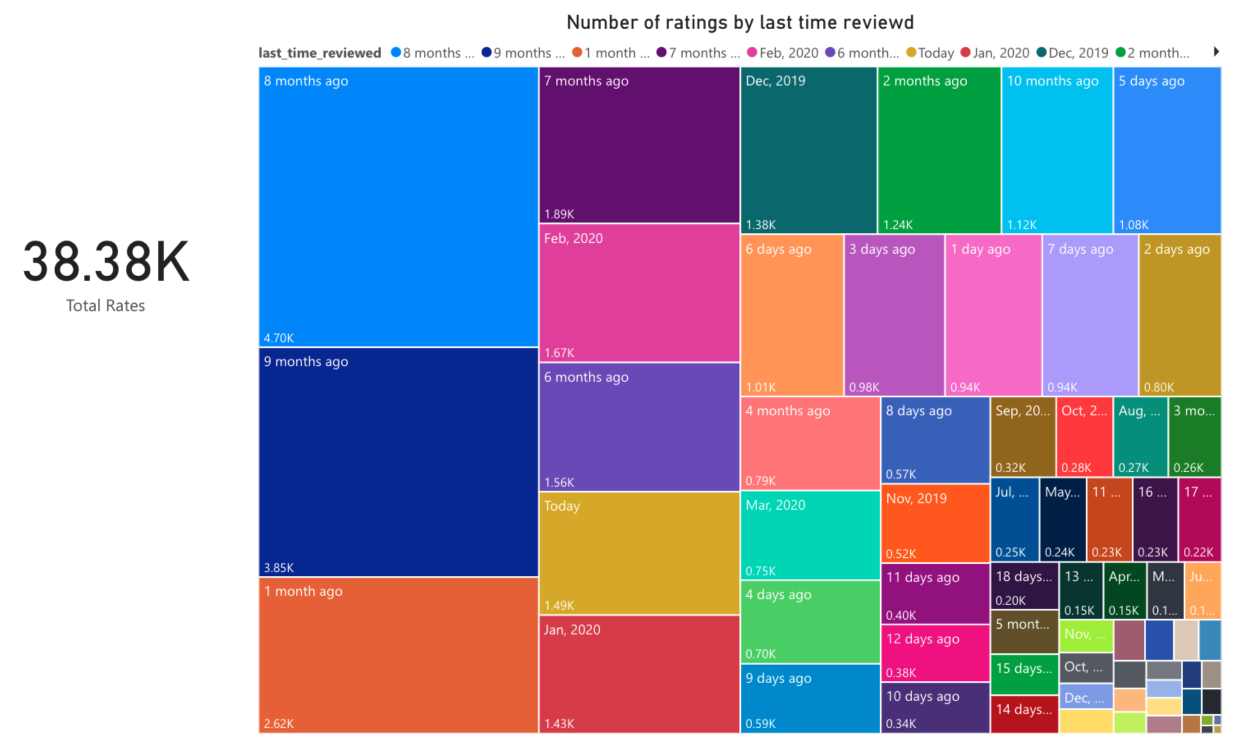


Figure 5: Number of ratings by last time reviewed.

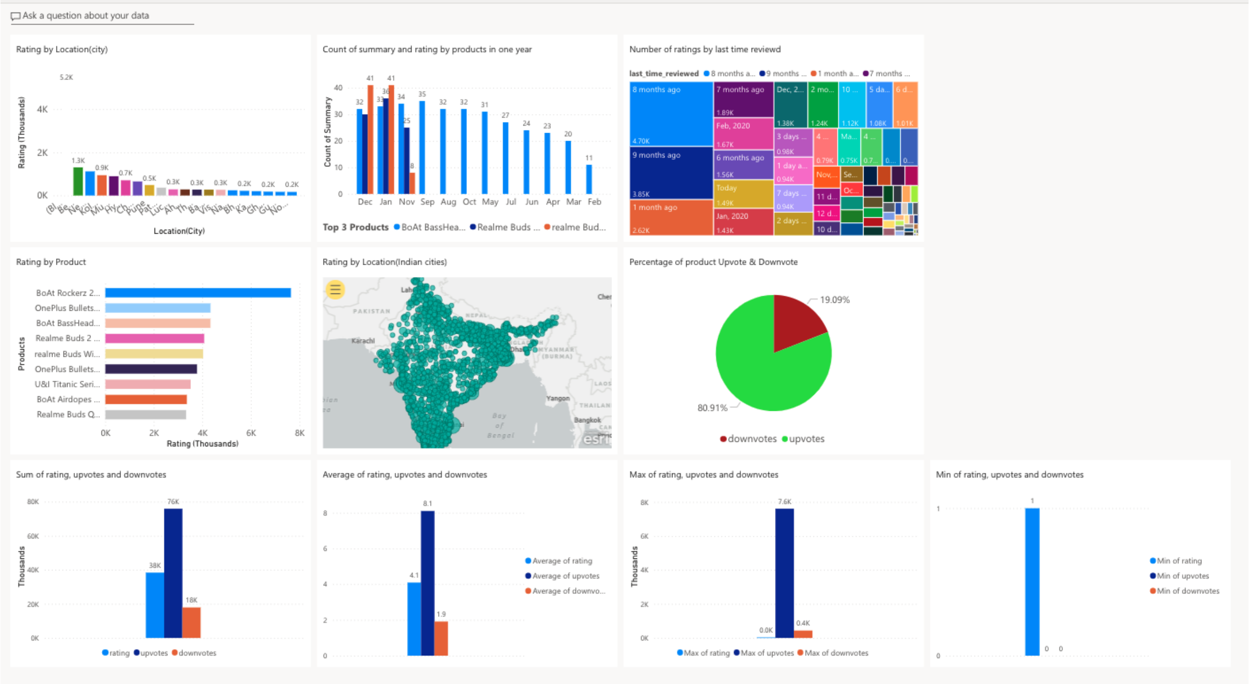


Figure 6: Dashboard: Flipkart products review in India.

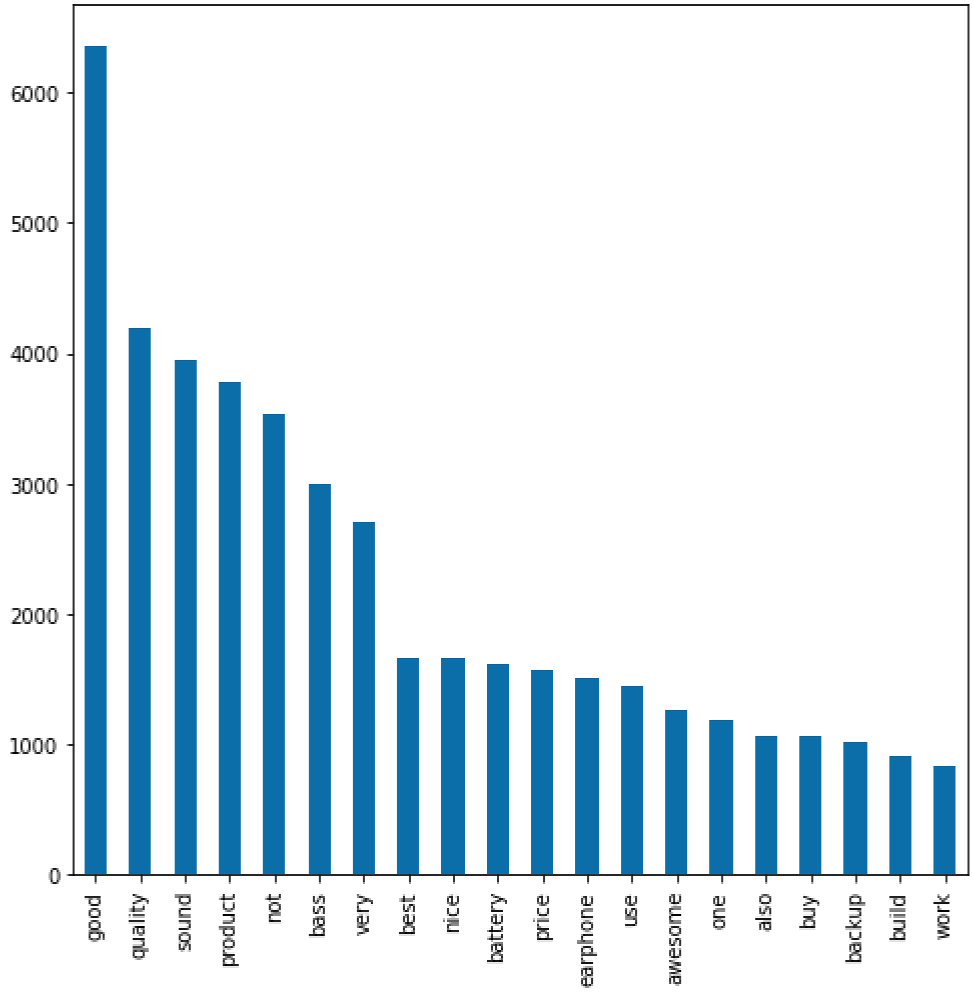


Figure 7: Count of all words

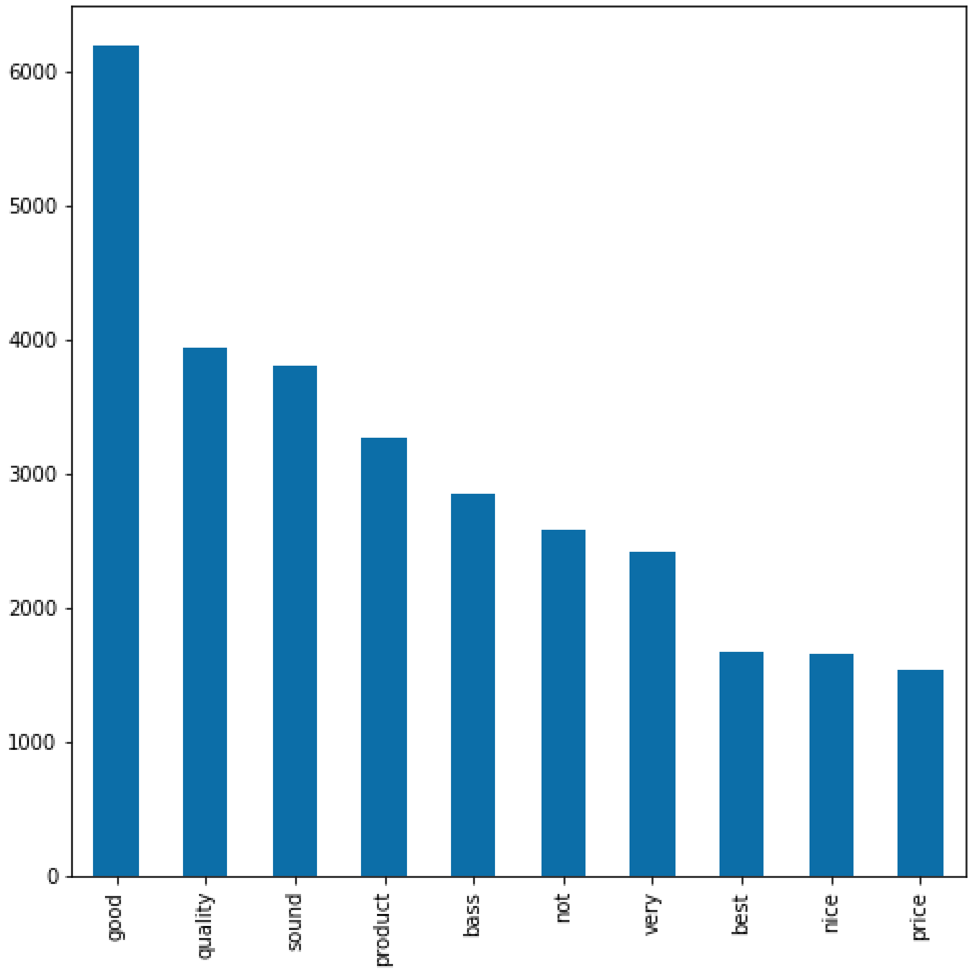


Figure 8: Count of positive words

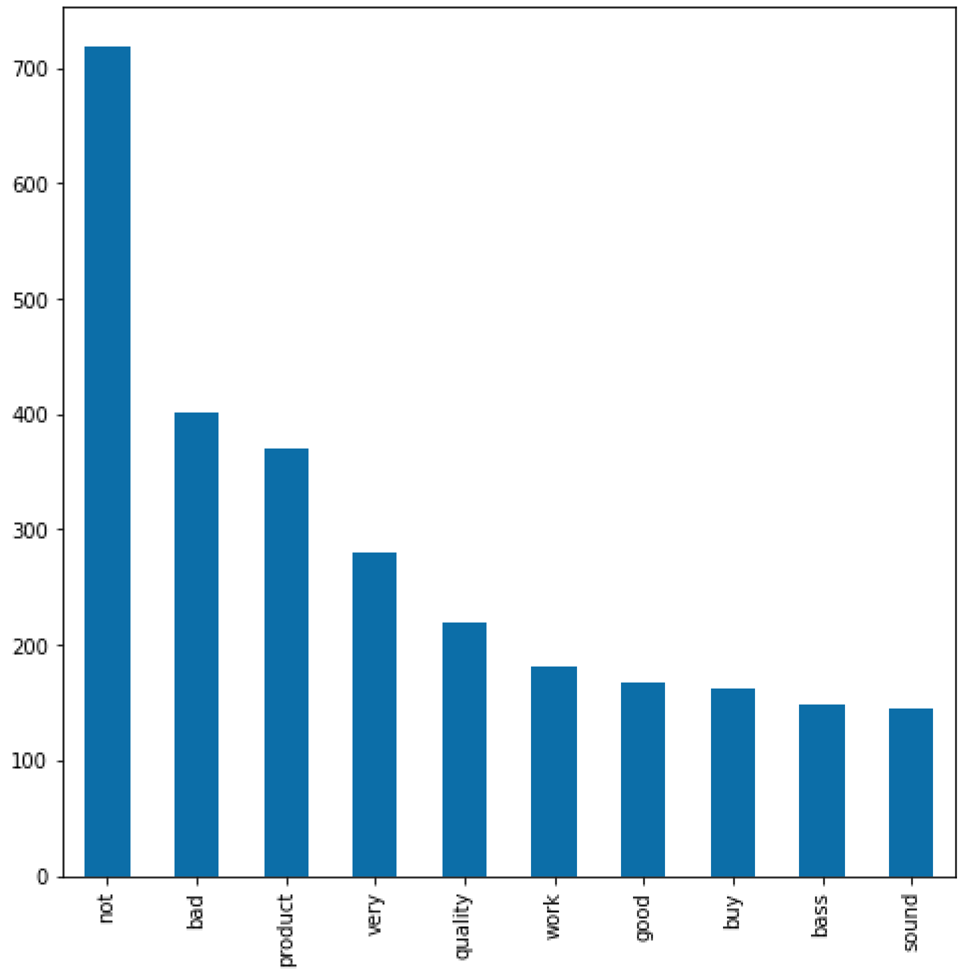


Figure 9: Count of Negative words

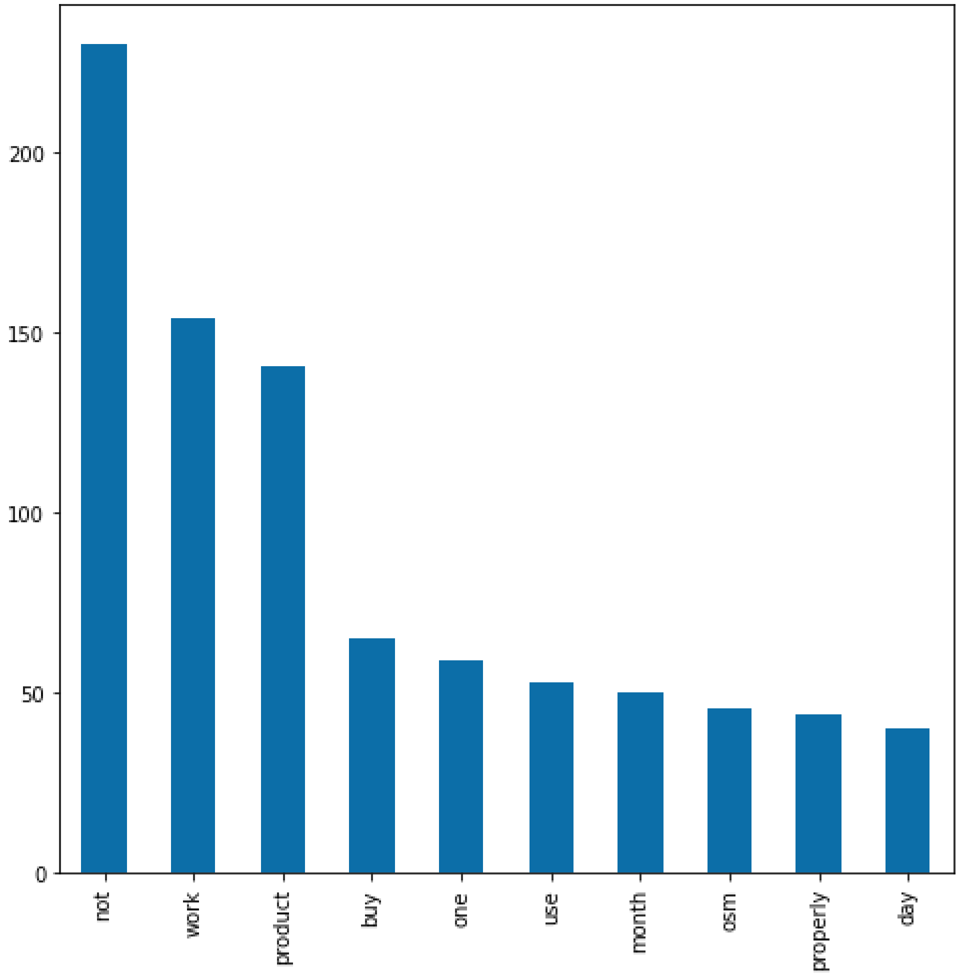
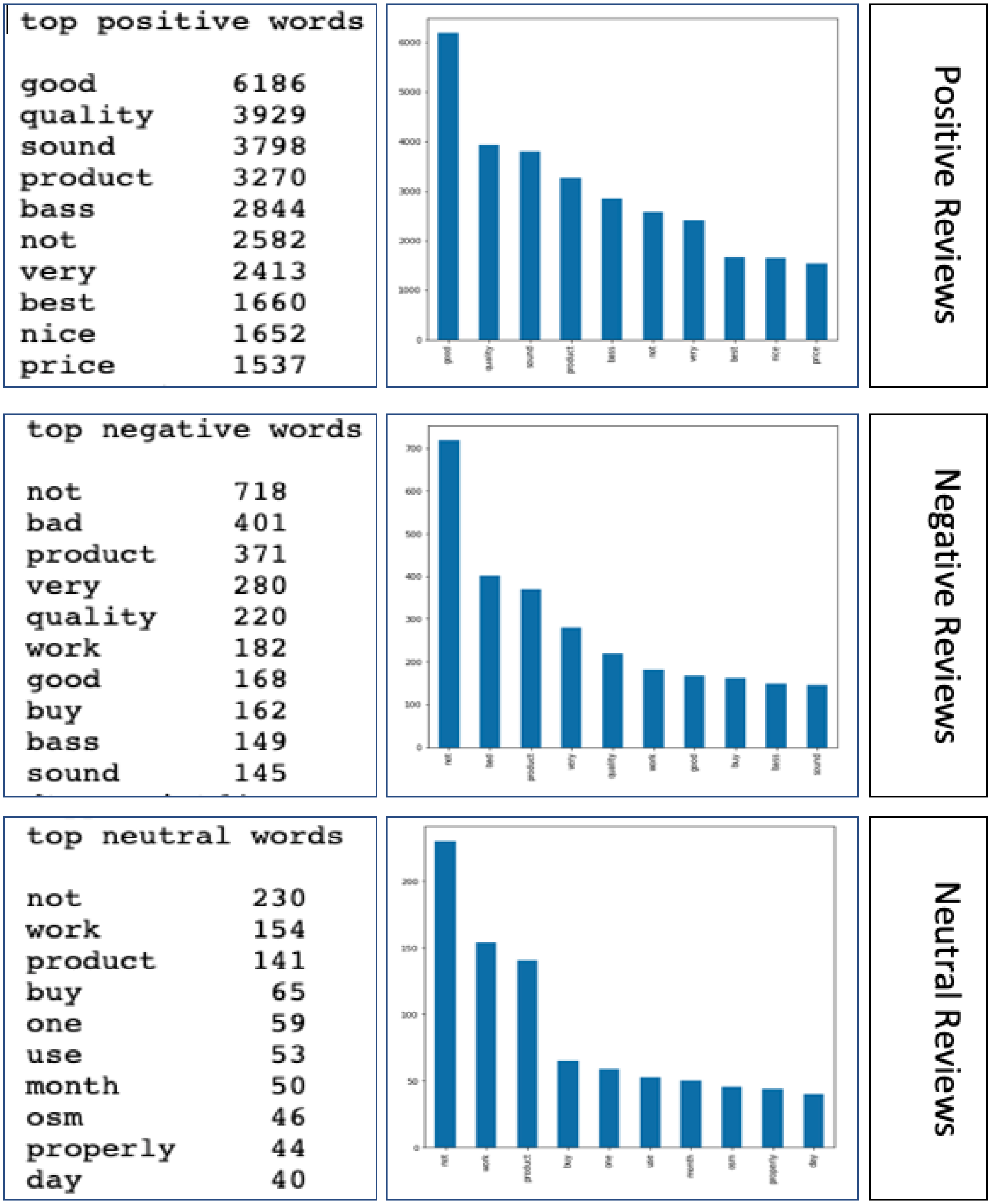


Figure 10: Count of Neutral words

Figure#11: Dashboard of Reviews

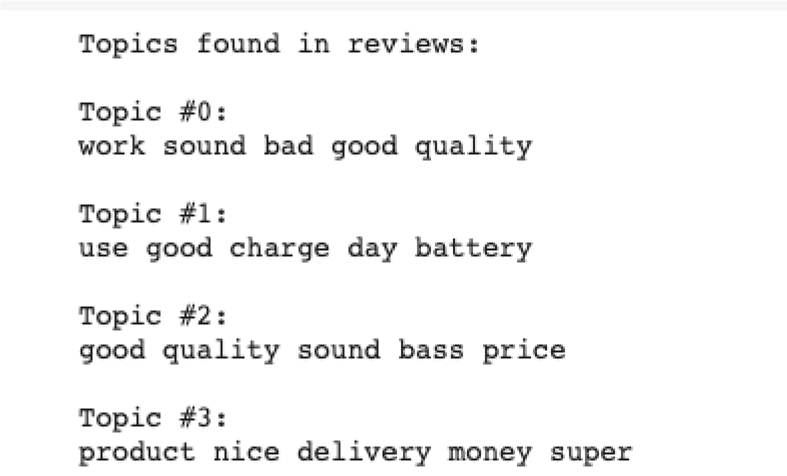


```
top positive words

good          6186
quality       3929
sound         3798
product       3270
bass          2844
not           2582
very          2413
best          1660
nice          1652
price         1537
```

```
top negative words

not           718
bad           401
product       371
very          280
quality       220
work          182
good          168
buy           162
bass          149
sound         145
```

```
top neutral words

not            230
work           154
product        141
buy             65
one             59
use             53
month           50
osm             46
properly        44
day             40
```

Figure 11: Dashboard of Positive, Negative and Neutral Words (Reviews)



```
Topics found in reviews:

Topic #0:
work sound bad good quality

Topic #1:
use good charge day battery

Topic #2:
good quality sound bass price

Topic #3:
product nice delivery money super
```

Figure 12: Four Topics via LDA Model