# SEPHORA

# Project Final Report

BANA 295: Big Data Management

**Professor:**

Vagelis Hristidis

**UCI Student Team:**

Xiwen Li, Ye Xiao, Kun Zhang

June 10, 2018

# Content

## Preface

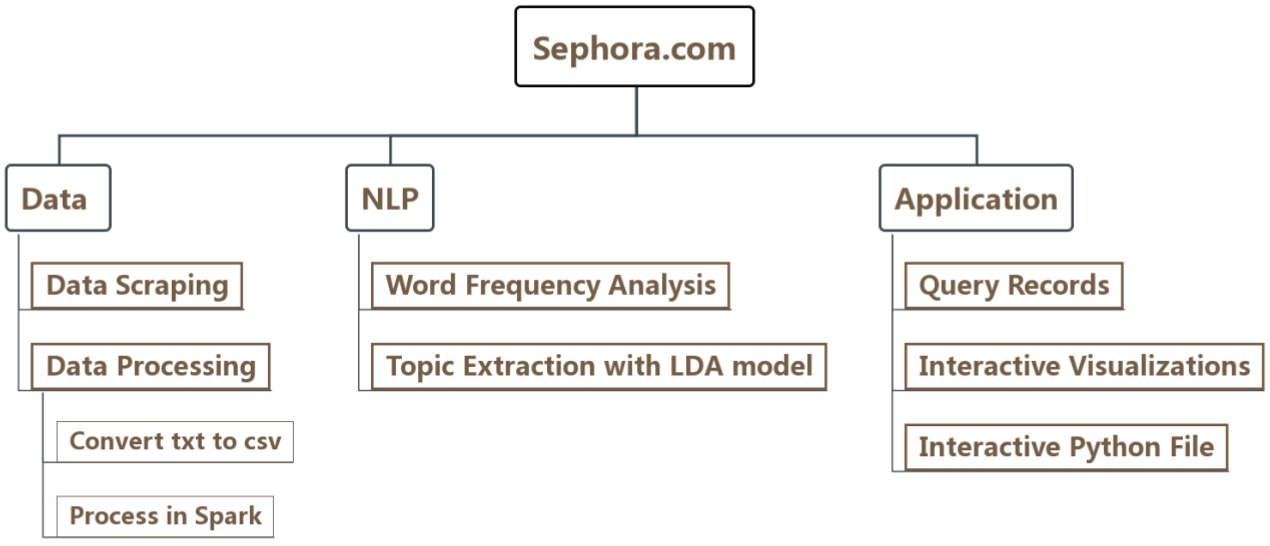


Figure.1 The overall figure of our project

# 1. Introduction

## 1.1 Background

Sephora is a global cosmetic retailer founded in 1970. Sephora carries more than 300 brands, and it operates approximately 2,300 stores in 33 countries worldwide, with over 430 stores across the Americas. Sephora provides an overall rating for every single product it carries as well as a brief specification, but it is important to read through the reviews to see other user's opinion in words before purchasing the products because a high rating does not suggest if the product is appropriate for you. Also, the product specification never mentions the negative side of the product. For example, a cleanser may have a high rating because it has a strong cleaning power, but you happen to have a sensitive skin and you need a gentle cleanser. In this case, if you only look at the overall rating of the product, it could be misleading.

## 1.2 Project Objectives

Our team thinks that customer reviews can be tremendously helpful since they contain first-hand experience from a wide range of people. However, a lot of products on Sephora.com have hundreds or even thousands of reviews. It is excruciating to read all of them. In order to extract valuable information from reviews, we summarize reviews using Natural Language Processing. At the end, we are able to provide customers with keywords generalized form genuine feedbacks from old customers. Moreover, with the immense information we have, we are able to predict new customers' tastes based on their attributes like age, skin type, and so on. An easy application will be developed to expedite the process.

# 2.Data

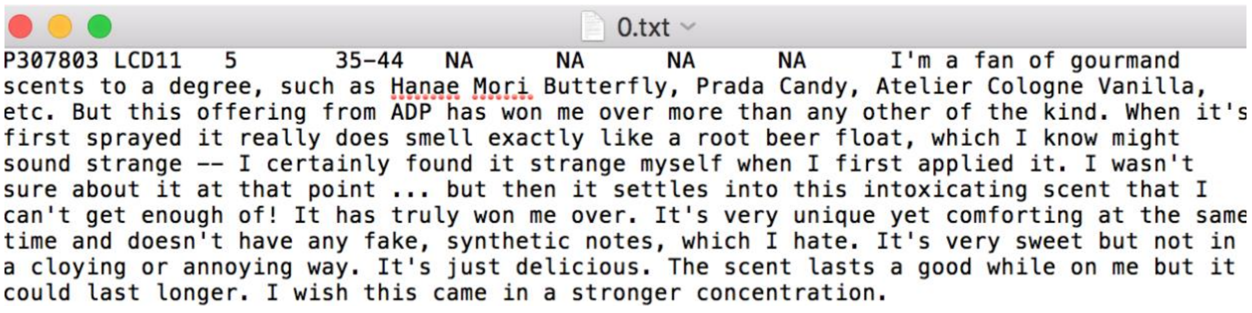
## 2.1 Data Scraping

First, I went to Sephora's brands page (https://www.sephora.com/brand/list.jsp) and scraped all the brand names. Then I pasted each brand names after (www.sephora.com/), so I got all the links of each brand's homepage. Secondly, from each brands' homepage source codes, I got the unique brand-ID corresponding to each brand. Thirdly, after clicking on the "see all" from brand's homepage, I inspected the page and found out there were a JSON data file. Then I found out the API which could return all the product-ID from brand-ID, so I scraped all the product-IDs from all the brand-IDs. Finally, from the product page, I inspected out that an API which could return all the reviews along with reviewer information of a product, so I used the API by automatic adjusting parameters to acquire all the reviews of all the products from Sephora. In addition, in order to make a precise analysis, I export 2 csv files during data scraping. Brands_URL.csv contains mapping relationship between brand-ID and its url, and Products_info.csv contains each product's name product-ID along with the brand name and brand-ID.

Detailed information about codes for data scraping can be found in Appendix B.

## 2.2 Data Description

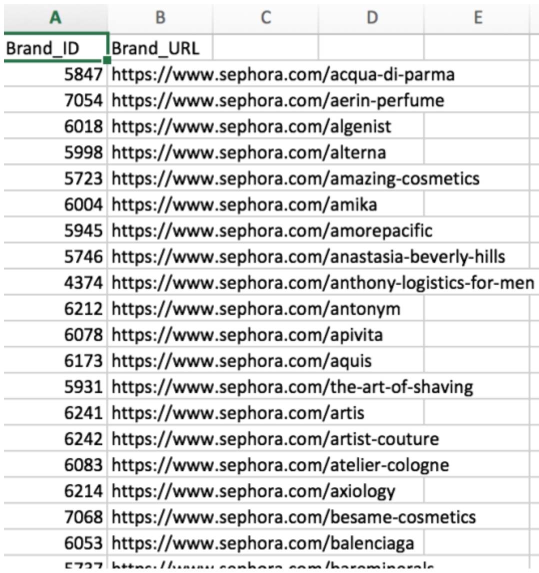
There are totally three data sets. Firstly, I scraped total of 235900 reviews from 312 brands' 9123 products. And Figure.2 is a screenshot of one review I scraped. Each record has a product-ID, a user-nickname, a rating, an age, an eye-color, a skin-tone, a skin-type, and the review. And if the reviewer left a section blank, it would show NA.



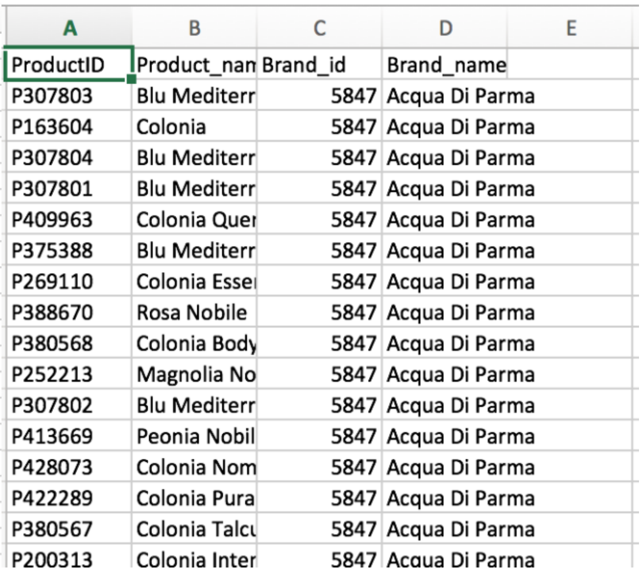
```
P307803 LCD11   5       35-44   NA      NA      NA      NA      I'm a fan of gourmand
scents to a degree, such as Hanae Mori Butterfly, Prada Candy, Atelier Cologne Vanilla,
etc. But this offering from ADP has won me over more than any other of the kind. When it's
first sprayed it really does smell exactly like a root beer float, which I know might
sound strange -- I certainly found it strange myself when I first applied it. I wasn't
sure about it at that point ... but then it settles into this intoxicating scent that I
can't get enough of! It has truly won me over. It's very unique yet comforting at the same
time and doesn't have any fake, synthetic notes, which I hate. It's very sweet but not in
a cloying or annoying way. It's just delicious. The scent lasts a good while on me but it
could last longer. I wish this came in a stronger concentration.
```

Figure.2 The screenshot of an example review file

And the second and third are about the information of brands and products. Figrue.3 is the screenshots of these two csv files (foreign key: Brand_ID).



| Brand_ID | Brand_URL |
|---|---|
| 5847 | https://www.sephora.com/acqua-di-parma |
| 7054 | https://www.sephora.com/aerin-perfume |
| 6018 | https://www.sephora.com/algenist |
| 5998 | https://www.sephora.com/alterna |
| 5723 | https://www.sephora.com/amazing-cosmetics |
| 6004 | https://www.sephora.com/amika |
| 5945 | https://www.sephora.com/amorepacific |
| 5746 | https://www.sephora.com/anastasia-beverly-hills |
| 4374 | https://www.sephora.com/anthony-logistics-for-men |
| 6212 | https://www.sephora.com/antonym |
| 6078 | https://www.sephora.com/apivita |
| 6173 | https://www.sephora.com/aquis |
| 5931 | https://www.sephora.com/the-art-of-shaving |
| 6241 | https://www.sephora.com/artis |
| 6242 | https://www.sephora.com/artist-couture |
| 6083 | https://www.sephora.com/atelier-cologne |
| 6214 | https://www.sephora.com/axiology |
| 7068 | https://www.sephora.com/besame-cosmetics |
| 6053 | https://www.sephora.com/balenciaga |
| 5737 | https://www.sephora.com/bareminerals |



| ProductID | Product_nam | Brand_id | Brand_name |
|---|---|---|---|
| P307803 | Blu Mediterr | 5847 | Acqua Di Parma |
| P163604 | Colonia | 5847 | Acqua Di Parma |
| P307804 | Blu Mediterr | 5847 | Acqua Di Parma |
| P307801 | Blu Mediterr | 5847 | Acqua Di Parma |
| P409963 | Colonia Quer | 5847 | Acqua Di Parma |
| P375388 | Blu Mediterr | 5847 | Acqua Di Parma |
| P269110 | Colonia Esse | 5847 | Acqua Di Parma |
| P388670 | Rosa Nobile | 5847 | Acqua Di Parma |
| P380568 | Colonia Body | 5847 | Acqua Di Parma |
| P252213 | Magnolia No | 5847 | Acqua Di Parma |
| P307802 | Blu Mediterr | 5847 | Acqua Di Parma |
| P413669 | Peonia Nobil | 5847 | Acqua Di Parma |
| P428073 | Colonia Nom | 5847 | Acqua Di Parma |
| P422289 | Colonia Pura | 5847 | Acqua Di Parma |
| P380567 | Colonia Talcu | 5847 | Acqua Di Parma |
| P200313 | Colonia Inter | 5847 | Acqua Di Parma |

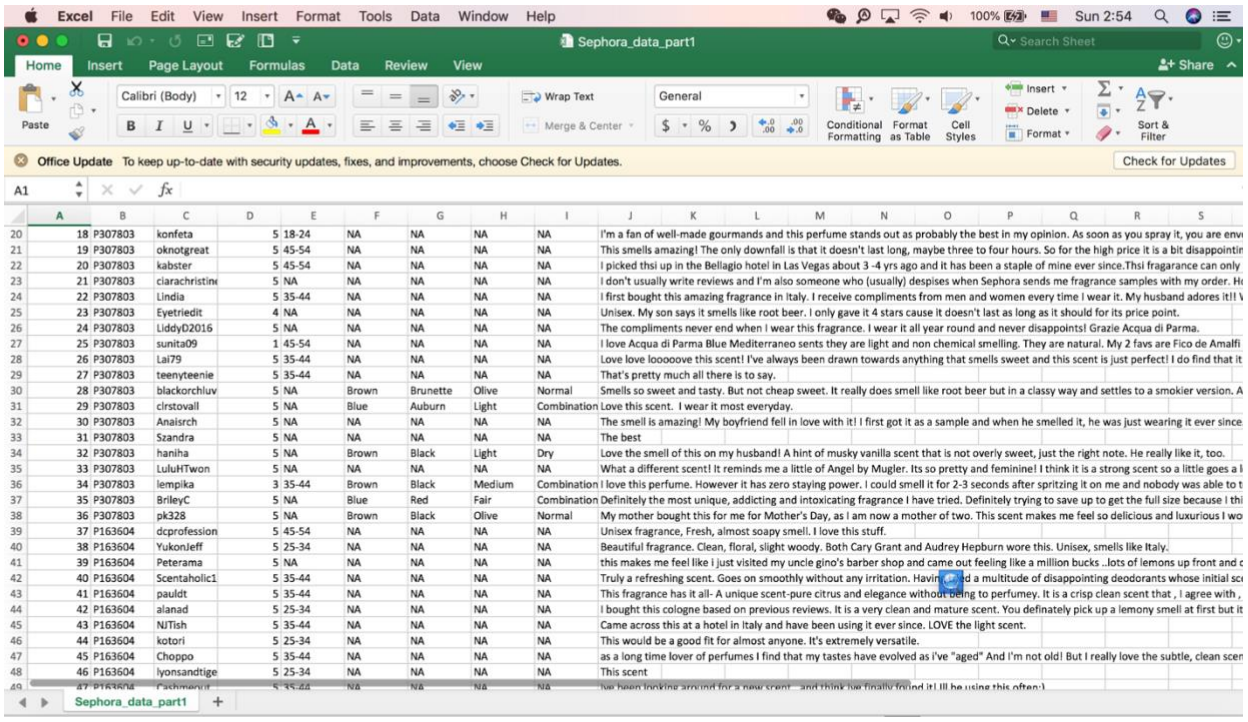
Figure.3 The screenshots of the data sets of brands and products

## 2.3 Data Processing

### 2.3.1 Converting txt to csv

In order to make the further analysis easier, I convert the txt reviews files to csv format. And Figure.4 is the screenshot of the csv generated.

Detailed information about how to convert the txt into csv is shown in Appendix C.



Excel File Edit View Insert Format Tools Data Window Help
Sephora_data_part1
Home Insert Page Layout Formulas Data Review View
Calibri (Body) 12
Wrap Text
Merge & Center
General
Conditional Formatting
Format as Table
Cell Styles
Insert
Delete
Format
Sort & Filter
Office Update To keep up-to-date with security updates, fixes, and improvements, choose Check for Updates.
Check for Updates
A1
Sephora_data_part1

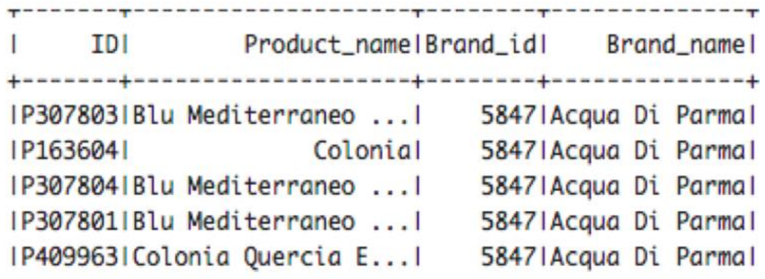
| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 20 | 18 | P307803 | konfeta | 5 | 18-24 | NA | NA | NA | NA | I'm a fan of well-made gourmands and this perfume stands out as probably the best in my opinion. As soon as you spray it, you are env |
| 21 | 19 | P307803 | oknotgreat | 5 | 45-54 | NA | NA | NA | NA | This smells amazing! The only downfall is that it doesn't last long, maybe three to four hours. So for the high price it is a bit disappointin |
| 22 | 20 | P307803 | kabster | 5 | 45-54 | NA | NA | NA | NA | I picked thsi up in the Bellagio hotel in Las Vegas about 3 -4 yrs ago and it has been a staple of mine ever since.Thsi fragarance can only |
| 23 | 21 | P307803 | ciarachristin | 5 | NA | NA | NA | NA | NA | I don't usually write reviews and I'm also someone who (usually) despises when Sephora sends me fragrance samples with my order. H |
| 24 | 22 | P307803 | Lindia | 5 | 35-44 | NA | NA | NA | NA | I first bought this amazing fragrance in Italy. I receive compliments from men and women every time I wear it. My husband adores it!! |
| 25 | 23 | P307803 | Eyetriedit | 4 | NA | NA | NA | NA | NA | Unisex. My son says it smells like root beer. I only gave it 4 stars cause it doesn't last as long as it should for its price point. |
| 26 | 24 | P307803 | LiddyD2016 | 5 | NA | NA | NA | NA | NA | The compliments never end when I wear this fragrance. I wear it all year round and never disappoints! Grazie Acqua di Parma. |
| 27 | 25 | P307803 | sunita09 | 1 | 45-54 | NA | NA | NA | NA | I love Acqua di Parma Blue Mediterraneo sents they are light and non chemical smelling. They are natural. My 2 favs are Fico de Amalfi |
| 28 | 26 | P307803 | Lai79 | 5 | 35-44 | NA | NA | NA | NA | Love love looooove this scent! I've always been drawn towards anything that smells sweet and this scent is just perfect! I do find that it |
| 29 | 27 | P307803 | teenyteenie | 5 | 35-44 | NA | NA | NA | NA | That's pretty much all there is to say. |
| 30 | 28 | P307803 | blackorchluv | 5 | NA | Brown | Brunette | Olive | Normal | Smells so sweet and tasty. But not cheap sweet. It really does smell like root beer but in a classy way and settles to a smokier version. A |
| 31 | 29 | P307803 | clrstovall | 5 | NA | Blue | Auburn | Light | Combination | Love this scent.  I wear it most everyday. |
| 32 | 30 | P307803 | Anaisrch | 5 | NA | NA | NA | NA | NA | The smell is amazing! My boyfriend fell in love with it! I first got it as a sample and when he smelled it, he was just wearing it ever since |
| 33 | 31 | P307803 | Szandra | 5 | NA | NA | NA | NA | NA | The best |
| 34 | 32 | P307803 | haniha | 5 | NA | Brown | Black | Light | Dry | Love the smell of this on my husband! A hint of musky vanilla scent that is not overly sweet, just the right note. He really like it, too. |
| 35 | 33 | P307803 | LuluHTwon | 5 | NA | NA | NA | NA | NA | What a different scent! It reminds me a little of Angel by Mugler. Its so pretty and feminine! I think it is a strong scent so a little goes a l |
| 36 | 34 | P307803 | lempika | 3 | 35-44 | Brown | Black | Medium | Combination | I love this perfume. However it has zero staying power. I could smell it for 2-3 seconds after spritzing it on me and nobody was able to t |
| 37 | 35 | P307803 | BrileyC | 5 | NA | Blue | Red | Fair | Combination | Definitely the most unique, addicting and intoxicating fragrance I have tried. Definitely trying to save up to get the full size because I thi |
| 38 | 36 | P307803 | pk328 | 5 | NA | Brown | Black | Olive | Normal | My mother bought this for me for Mother's Day, as I am now a mother of two. This scent makes me feel so delicious and luxurious I wo |
| 39 | 37 | P163604 | dcprofession | 5 | 45-54 | NA | NA | NA | NA | Unisex fragrance, Fresh, almost soapy smell. I love this stuff. |
| 40 | 38 | P163604 | YukonJeff | 5 | 25-34 | NA | NA | NA | NA | Beautiful fragrance. Clean, floral, slight woody. Both Cary Grant and Audrey Hepburn wore this. Unisex, smells like Italy. |
| 41 | 39 | P163604 | Peterama | 5 | NA | NA | NA | NA | NA | this makes me feel like i just visited my uncle gino's barber shop and came out feeling like a million bucks ..lots of lemons up front and c |
| 42 | 40 | P163604 | Scentaholic1 | 5 | 35-44 | NA | NA | NA | NA | Truly a refreshing scent. Goes on smoothly without any irritation. Havin[illegible]d a multitude of disappointing deodorants whose initial sc |
| 43 | 41 | P163604 | pauldt | 5 | 35-44 | NA | NA | NA | NA | This fragrance has it all- A unique scent-pure citrus and elegance without being to perfumey. It is a crisp clean scent that , I agree with , |
| 44 | 42 | P163604 | alanad | 5 | 25-34 | NA | NA | NA | NA | I bought this cologne based on previous reviews. It is a very clean and mature scent. You definately pick up a lemony smell at first but it |
| 45 | 43 | P163604 | NJTish | 5 | 35-44 | NA | NA | NA | NA | Came across this at a hotel in Italy and have been using it ever since. LOVE the light scent. |
| 46 | 44 | P163604 | kotori | 5 | 25-34 | NA | NA | NA | NA | This would be a good fit for almost anyone. It's extremely versatile. |
| 47 | 45 | P163604 | Choppo | 5 | 35-44 | NA | NA | NA | NA | as a long time lover of perfumes I find that my tastes have evolved as i've "aged" And I'm not old! But I really love the subtle, clean scen |
| 48 | 46 | P163604 | lyonsandtige | 5 | 25-34 | NA | NA | NA | NA | This scent |
| 49 | 47 | P163604 | Cashmeout | 5 | 35-44 | NA | NA | NA | NA | Ive been looking around for a new scent , and think Ive finally found it! Ill be using this often:) |

Figure.4 Screenshot of the csv file converted from txt

### 2.3.2 Processing data through Spark

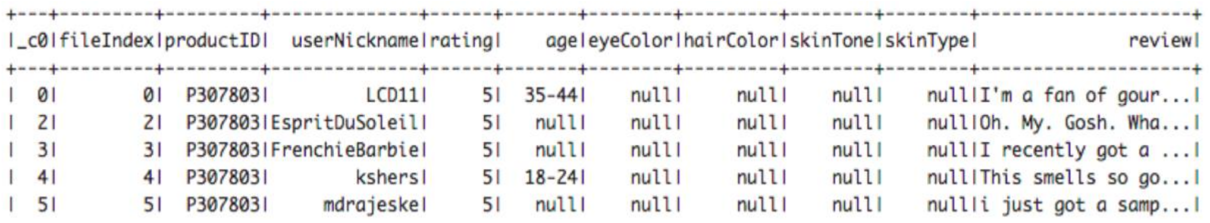
After crawling data from Sephora website, we found that the size of data is around 10GB which meet the requirements for this project. Simultaneously, another problem is that the data is so large that it is hard to process it locally. So, we decided to upload it to AWS S3 and process it remotely using spark.

After uploading data to S3, we created a spark cluster and used Zeppelin as a notepad to write and execute code for big data processing purpose. Firstly, we used command “spark.read” to read data and create RDD “review” and “product”. We could also use review.show() as an action command in spark to view top 20 rows in each RDD.



```
+-------+--------------------+--------+--------------+
|     ID|        Product_name|Brand_id|    Brand_name|
+-------+--------------------+--------+--------------+
|P307803|Blu Mediterraneo ...|    5847|Acqua Di Parma|
|P163604|             Colonia|    5847|Acqua Di Parma|
|P307804|Blu Mediterraneo ...|    5847|Acqua Di Parma|
|P307801|Blu Mediterraneo ...|    5847|Acqua Di Parma|
|P409963|Colonia Quercia E...|    5847|Acqua Di Parma|
```

Figure.5 Sample data of “Product”

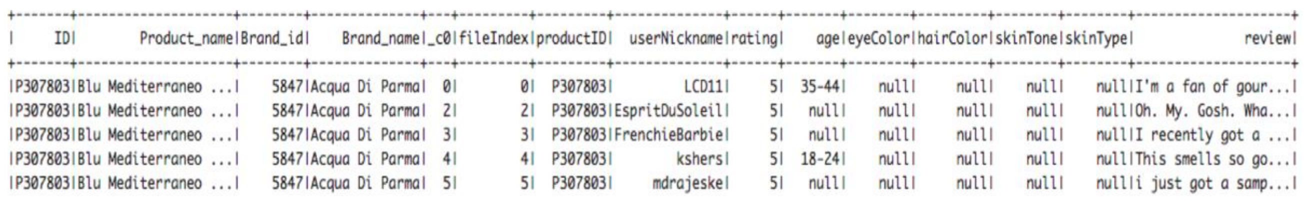


```
+---+---------+---------+--------------+------+------+--------+---------+--------+--------+--------------------+
|_c0|fileIndex|productID|  userNickname|rating|   age|eyeColor|hairColor|skinTone|skinType|              review|
+---+---------+---------+--------------+------+------+--------+---------+--------+--------+--------------------+
|  0|        0|  P307803|         LCD11|     5| 35-44|    null|     null|    null|    null|I'm a fan of gour...|
|  2|        2|  P307803|EspritDuSoleil|     5|  null|    null|     null|    null|    null|Oh. My. Gosh. Wha...|
|  3|        3|  P307803| FrenchieBarbie|     5|  null|    null|     null|    null|    null|I recently got a ...|
|  4|        4|  P307803|        kshers|     5| 18-24|    null|     null|    null|    null|This smells so go...|
|  5|        5|  P307803|     mdrajeske|     5|  null|    null|     null|    null|    null|i just got a samp...|
```

Figure.6 Sample data of “Review”

Like what the tables show here, from Figure 6 and Figure 7, we could know the demographic information of each customer like age, gender, skin type etc. Also, we could know customer reviews and which product they leave the review to.

In order to match the review data, customer information and product information together and do some processing work, we decided to use SparkSQL since the data is structured and a DataFrame provides more operations which are not available on RDDs like running SQL queries. So, we created 2 temp tables using “CreateOrReplaceTempView” command to execute SQL queries. By using the command below, we joined two tables together.



```
+-------+--------------------+--------+--------------+---+---------+---------+--------------+------+------+--------+---------+--------+--------+--------------------+
|     ID|        Product_name|Brand_id|    Brand_name|_c0|fileIndex|productID|  userNickname|rating|   age|eyeColor|hairColor|skinTone|skinType|              review|
+-------+--------------------+--------+--------------+---+---------+---------+--------------+------+------+--------+---------+--------+--------+--------------------+
|P307803|Blu Mediterraneo ...|    5847|Acqua Di Parma|  0|        0|  P307803|         LCD11|     5| 35-44|    null|     null|    null|    null|I'm a fan of gour...|
|P307803|Blu Mediterraneo ...|    5847|Acqua Di Parma|  2|        2|  P307803|EspritDuSoleil|     5|  null|    null|     null|    null|    null|Oh. My. Gosh. Wha...|
|P307803|Blu Mediterraneo ...|    5847|Acqua Di Parma|  3|        3|  P307803| FrenchieBarbie|     5|  null|    null|     null|    null|    null|I recently got a ...|
|P307803|Blu Mediterraneo ...|    5847|Acqua Di Parma|  4|        4|  P307803|        kshers|     5| 18-24|    null|     null|    null|    null|This smells so go...|
|P307803|Blu Mediterraneo ...|    5847|Acqua Di Parma|  5|        5|  P307803|     mdrajeske|     5|  null|    null|     null|    null|    null|i just got a samp...|
```

Figure.7  The structure of the whole table

After joining data together, we used command like "group by", "Sort" etc. under sqlContext to select data under specific constraints and filter data for analysis purpose.

Detailed codes for Spark process are shown in Appendix D.

# 3.Nature Language Process

## 3.1 Word Frequency Analysis

After descriptive analysis, we found that the product "Naked Palette Collection" is very popular which has more than 20,000 reviews. So, in order to dig the reason why people give it positive reviews and negative reviews, we filtered reviews for this product and define reviews with rating=1 as negative reviews and reviews with rating = 5 as positive reviews. We create an RDD for review data and store it in S3. Then, we imported data into Python and did word frequency analysis and word cloud chart for both positive reviews and negative reviews.

From the word frequency chart for positive reviews, we could see that words like "palette", "brush", "skin" appear frequently. We may simply infer that people like this eye shadow because it's brush is easy to use, or the eye shadow looks great on skin.

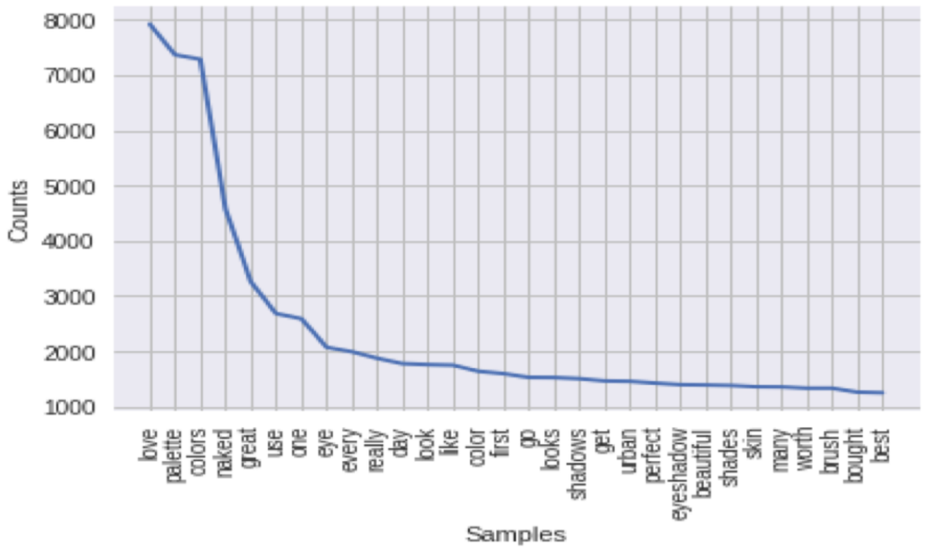


Figure.8 Word frequency for "positive reviews"

From the word frequency line chart for negative reviews, we could see that "colors" is the most frequent word so we may infer that people give it low rating because its color.

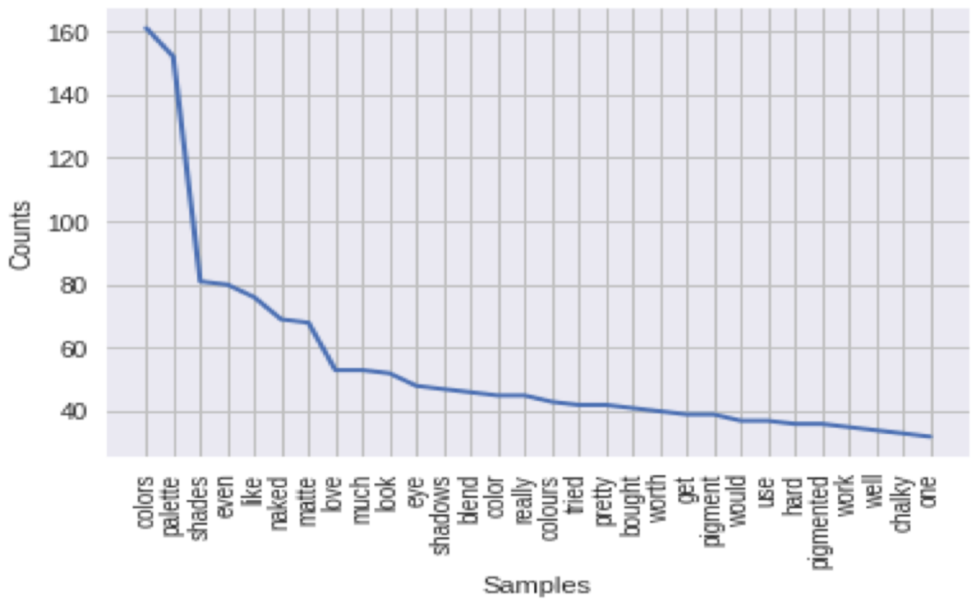


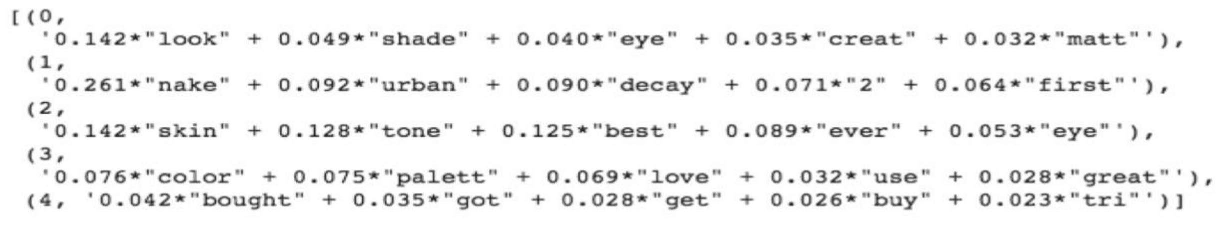
Figure.9 word frequency for “negative reviews”

Detailed codes for word frequency analysis are shown in Appendix E.

## 3.2 Topic Extraction with LDA model

In order to know more about the reasons why people like or dislike the product, we decided to use LDA model to extract topics from reviews.

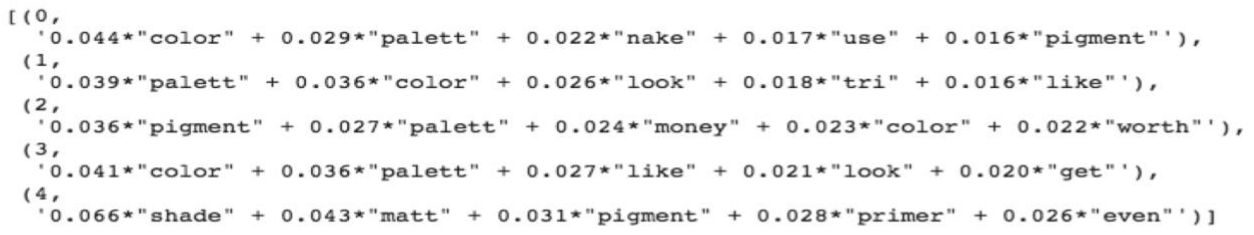
Firstly, we tokenized words from reviews with “RegexTokenizer” command. Then, since lots of words are stop words like “I”, “are”, “is” etc. which are meaningless for analysis purpose, we removed stop words from each review. Also, we changed all the words in to stem form since words always have different tense and will be recognized as different words if we didn’t stem them (e.g like vs likes). After that, we changed each review into bags of words and then combine all those reviews into a matrix so that it can be used for LDA model analysis.



```
[(0,
  '0.142*"look" + 0.049*"shade" + 0.040*"eye" + 0.035*"creat" + 0.032*"matt"'),
 (1,
  '0.261*"nake" + 0.092*"urban" + 0.090*"decay" + 0.071*"2" + 0.064*"first"'),
 (2,
  '0.142*"skin" + 0.128*"tone" + 0.125*"best" + 0.089*"ever" + 0.053*"eye"'),
 (3,
  '0.076*"color" + 0.075*"palett" + 0.069*"love" + 0.032*"use" + 0.028*"great"'),
 (4, '0.042*"bought" + 0.035*"got" + 0.028*"get" + 0.026*"buy" + 0.023*"tri"')]
```

Figure.10 Positive Reviews with Rating=5

Consequently, based on the result, we could infer that people love this eyeshadow firstly because it’s matt color; secondly, lots of people try it the first time and feels good so that they give it a high rating; additionally, some other people feel that the eye shadow matched greatly with their skin tone.



```
[(0,
  '0.044*"color" + 0.029*"palett" + 0.022*"nake" + 0.017*"use" + 0.016*"pigment"'),
 (1,
  '0.039*"palett" + 0.036*"color" + 0.026*"look" + 0.018*"tri" + 0.016*"like"'),
 (2,
  '0.036*"pigment" + 0.027*"palett" + 0.024*"money" + 0.023*"color" + 0.022*"worth"'),
 (3,
  '0.041*"color" + 0.036*"palett" + 0.027*"like" + 0.021*"look" + 0.020*"get"'),
 (4,
  '0.066*"shade" + 0.043*"matt" + 0.031*"pigment" + 0.028*"primer" + 0.026*"even"')]
```

Figure.11 Negative Reviews with Rating=1

As for the reasons why people give it negative reviews, some people don’t like its pigment color; also, some other consumers feel that this product is not worth the money and kind of expensive.

Detailed codes for LDA model are shown in Appendix F.

# 4. Application

## 4.1 Methodology

We have two datasets scripted from Sephora.com, one for all information of reviews (including age, rating, skin-type of reviewers, and review texts), and another for product information. So according to the product ID, we left joined the product information into review datasets. And the following operations are performed on this joined dataset.

### 4.1.1 Query Records

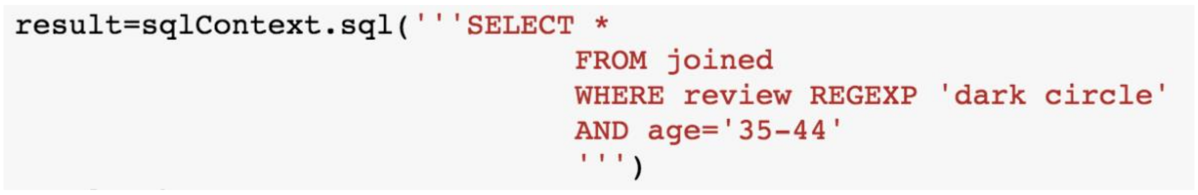
For millions of records, it’s time wasting to run filter function on local environment. So, we prefer to use Spark SQL to execute the filter process. We set six variables as customized inputs (Detailed introductions are shown in Table.1)

| Symbol | Explanation | Choices |
|---|---|---|
| -k | Keywords | Enter any keywords you are interested in |
| -a | Age ranges | '35-44', '18-24', '45-54', 'over 54', '25-34', '13-17' |
| -e | Eye colors | 'Brown', 'Blue', 'Hazel', 'Green', 'Gray' |
| -c | Hair colors | 'Brunette', 'Auburn', 'Black', 'Red', 'Gray', 'Blonde' |
| -t | Skin tones | 'Olive', 'Light', 'Medium', 'Fair', 'Porcelain', 'Deep', 'Tan', 'Dark', 'Ebony' |

|  |  |  |
|---|---|---|
| -s | Skin types | 'Normal', 'Combination', 'Dry', 'Oily' |

Table.1 The information of inputs

Based on the inputs, we can generate SQL syntaxes to select the fitted records. Below is a simple example in Figure.12.



```
result=sqlContext.sql('''SELECT *
                         FROM joined
                         WHERE review REGEXP 'dark circle'
                         AND age='35-44'
                         ''')
```

Figure.12 Example for selecting fitted records by Spark SQL.

In this example, we can select the reviews that contain 'dark circle' and the reviewer's age is in 35-44.

### 4.1.2 Interactive Visualizations

Once we have selected the relevant records, we want to show some interesting interactive visualizations for business insights. The main package used in python is plotly, which generates interactive graphs in html format (refer to Ye_SparkSQL.ipynb for original codes). For each query, it will automatically generate two pie charts for top 20 brands and products, one box plot of ratings for each brand, and one scatter plot for review texts.

### 4.1.3 Interactive Python File

We allow users to input parameters to generate graphs with the help of 'argparse' package (refer to Ye_SparkSQL.py for complete codes). The information of parameters is listed in Table.1. And the example of command prompt is,

```
python3 Ye_SparkSQL.py -k dark circle -a 35-44 -e Brown -c Brunette -t Olive
                              -s Normal
```

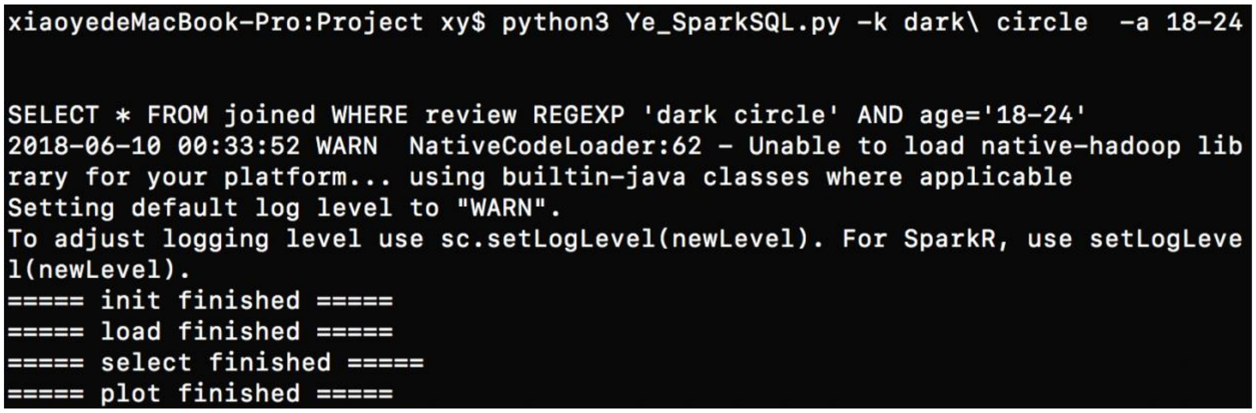
, where it searches the records that contain keywords 'dark circle', and that are written by the reviewers between 35 and 44 with brown eyes, brunette hairs, olive and normal skins.

## 4.2 Results

We use a simple example to better illustrate our works. When the command prompt is,

```
python3 Ye_SparkSQL.py -k dark\ circle  -a 18-24
```

, the terminal interface is shown in Figure.13.



```
xiaoyedeMacBook-Pro:Project xy$ python3 Ye_SparkSQL.py -k dark\ circle  -a 18-24


SELECT * FROM joined WHERE review REGEXP 'dark circle' AND age='18-24'
2018-06-10 00:33:52 WARN  NativeCodeLoader:62 - Unable to load native-hadoop lib
rary for your platform... using builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLeve
l(newLevel).
===== init finished =====
===== load finished =====
===== select finished =====
===== plot finished =====
```

Figure.13 The terminal interface after executing the example command prompt.

At the same time, it will output four graphs automatically (Shown in Figure.14, Figure.15, Figure.16, Figure.17. For integrated animations, refer to brands.html, products.html, boxplot of ratings.html, and scatter plot for reviews.html).

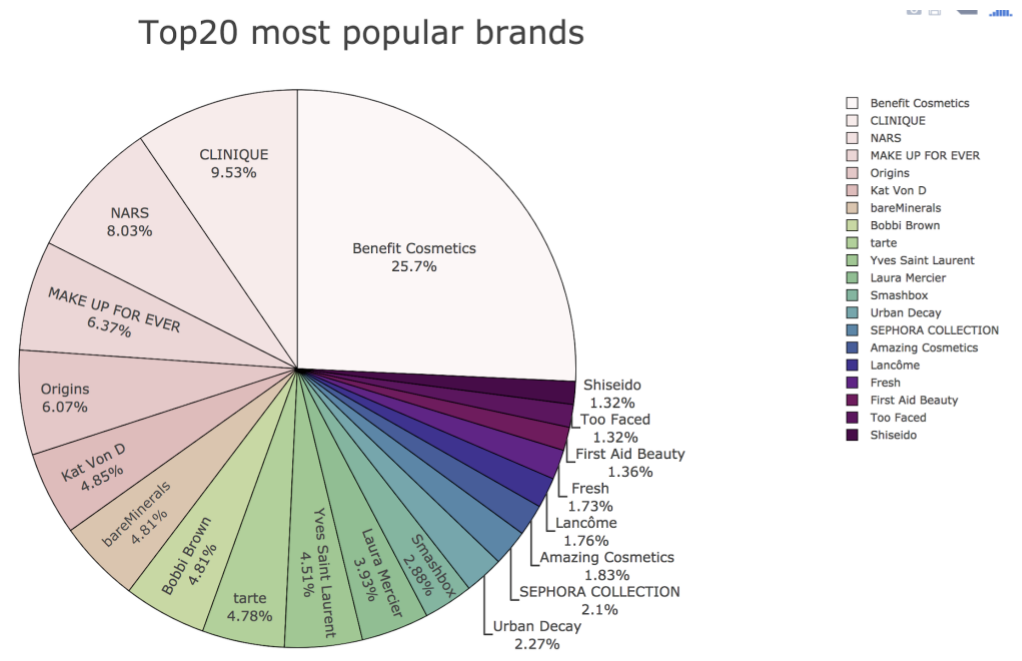


Figure.14 The top 20 most popular brands of the example command prompt

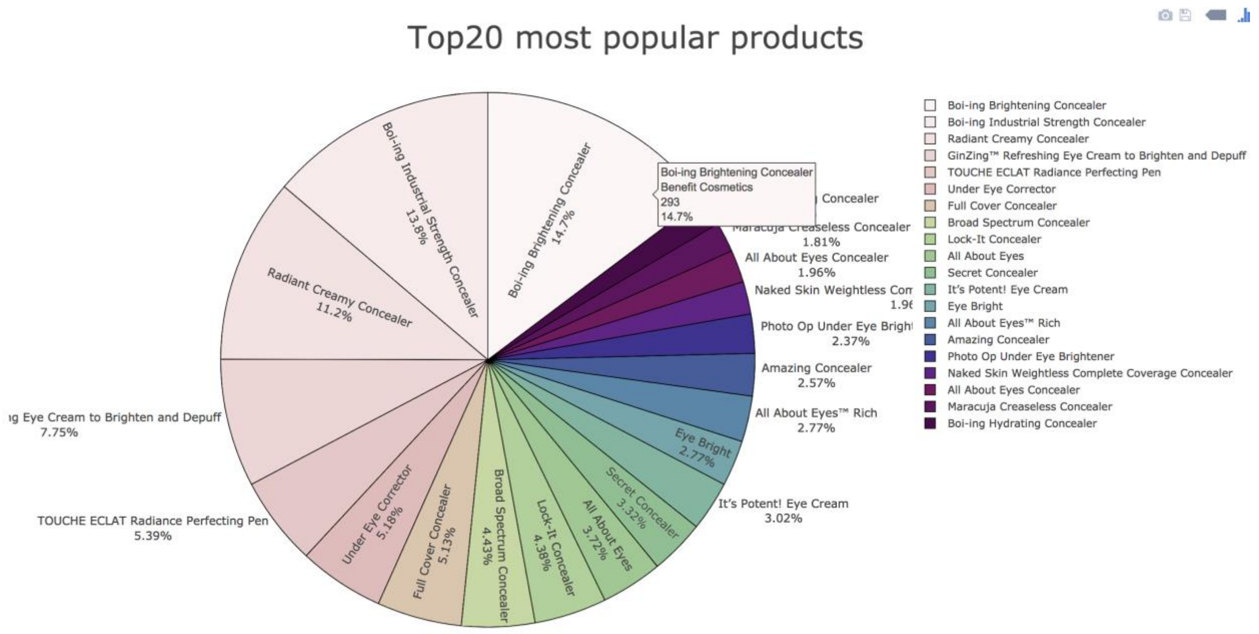


Figure.15 The top 20 most popular products of the example command prompt

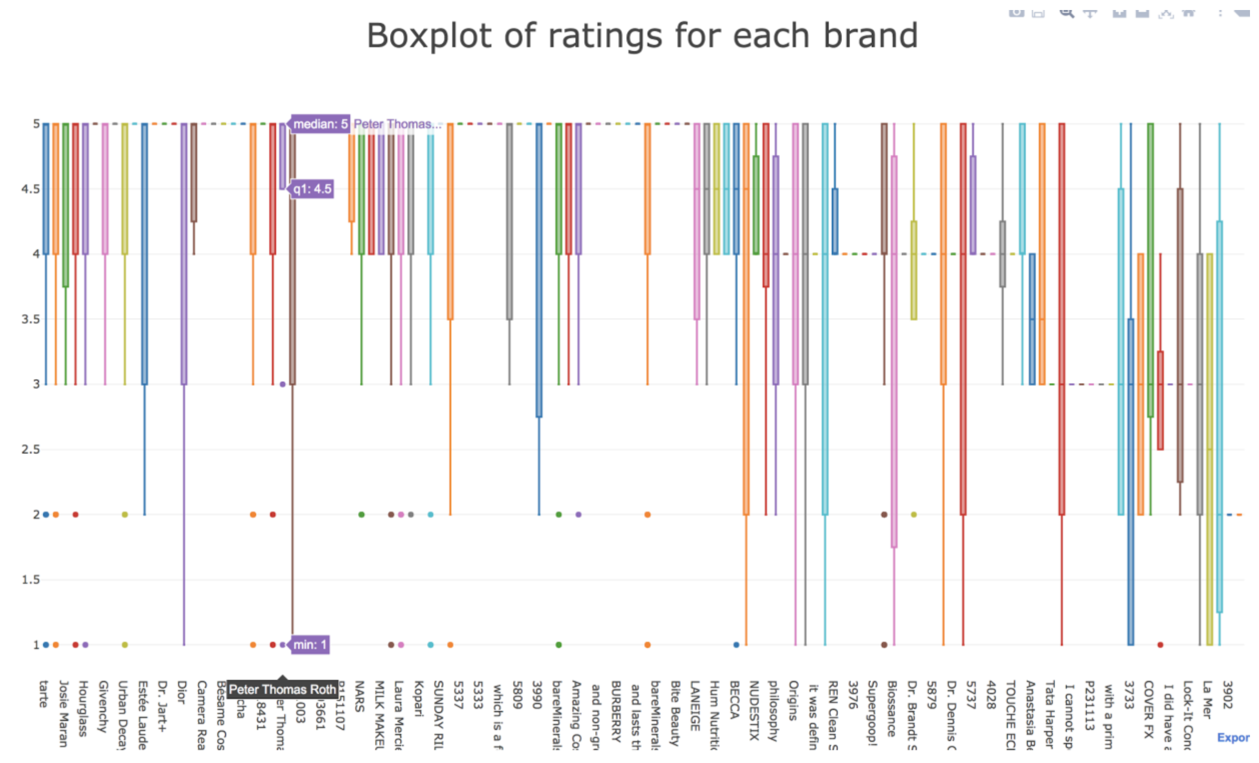


Figure.16 The box plot of ratings for each relevant brand

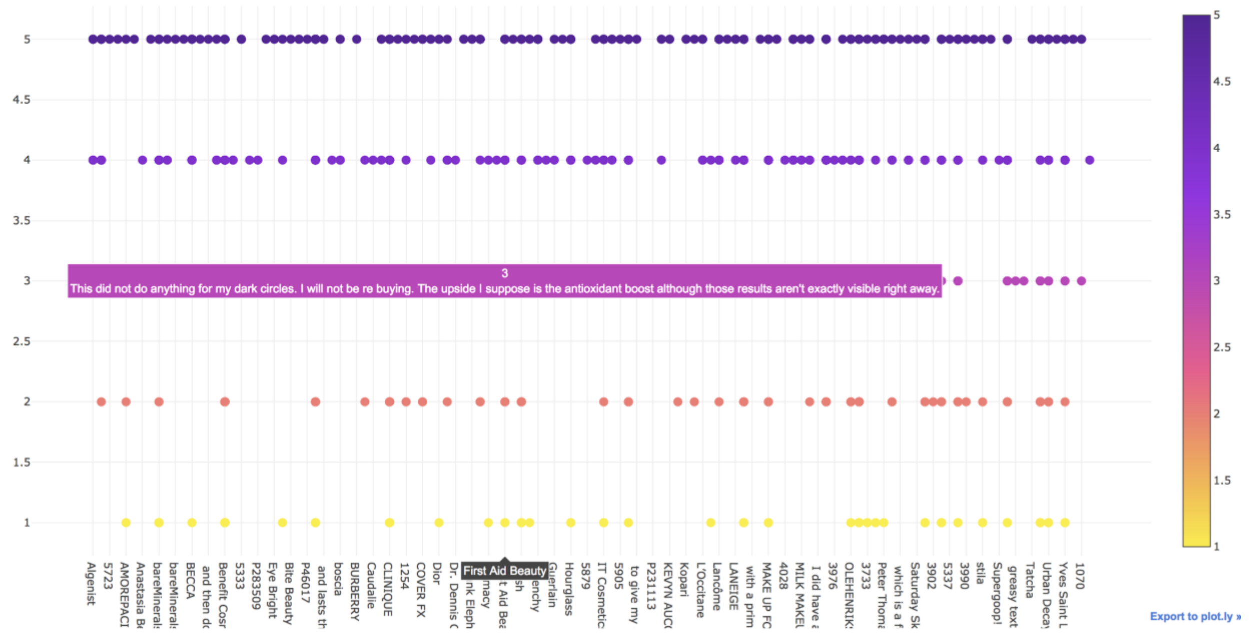

3
This did not do anything for my dark circles. I will not be re buying. The upside I suppose is the antioxidant boost although those results aren't exactly visible right away.

Figure.17 The scatter plot of relevant reviews

### 4.3 Limitations

The first obvious limitation is how to process the customized keywords. In general, people express the same meanings regardless of upper or lower case letters. But in our syntax, it simply uses regular expression to match the exact words. For example, we also input ‘dark circle’ as keywords, but it’s impossible to select the reviews that contain ‘Dark Circle’ or ‘Dark circle’. It is a distinct drawback and requires more considerations in the future.

## 5. Conclusion

regarding this project, we firstly crawled consumer and review data from Sephora website and processed it in Spark. After that, we did descriptive analysis for consumers’ demographic distribution and popularity of brands and products. Then, we selected one product called “Naked Palette Collection” and analyzed its positive and negative reviews in detail with NLP model so that the company could optimize based the reason why consumers dislike it and treat those reasons why people like it as selling point to do promotion. Finally, based on NLP processing, we build a recommendation system which could recommend most relevant brands and products if you enter some keywords. Also, the relevant reviews could also be found so that consumers could select products based those relevant reviews.

# Appendices

## A: Collaboration Details

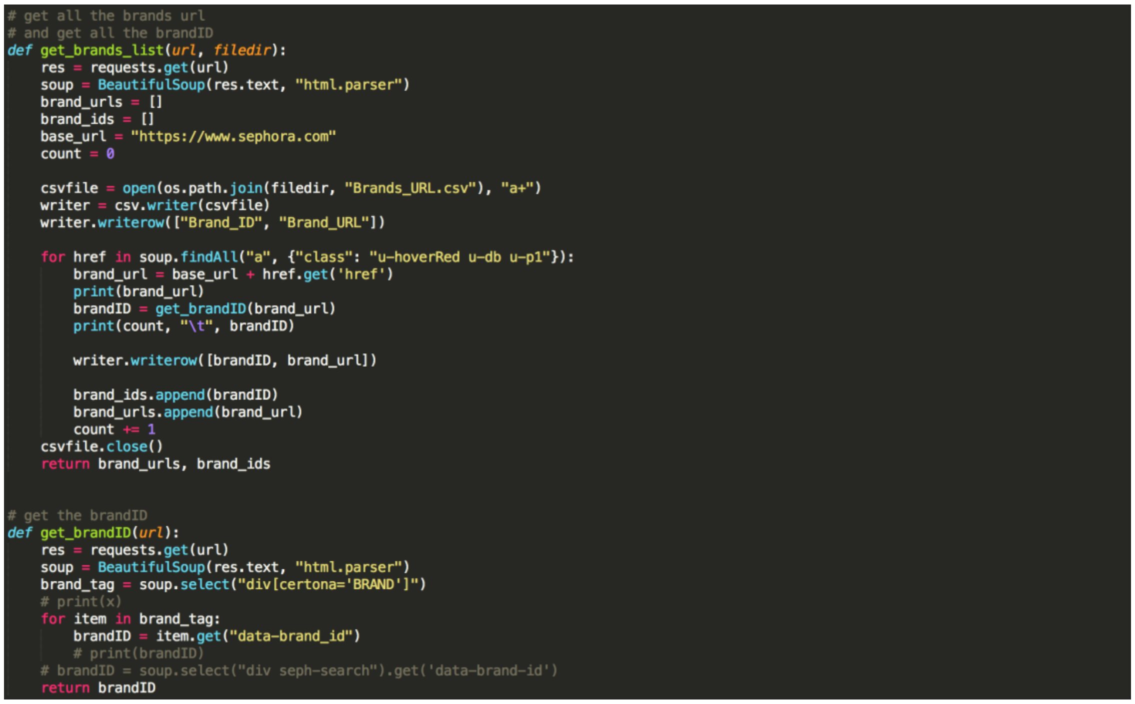
Our team collaborate well together, and we all agree that works distributes between three team members equally and fairly.

Xiwen Li: Data scraping, data processing, and PPT.

Ye Xiao: Application development, and PPT

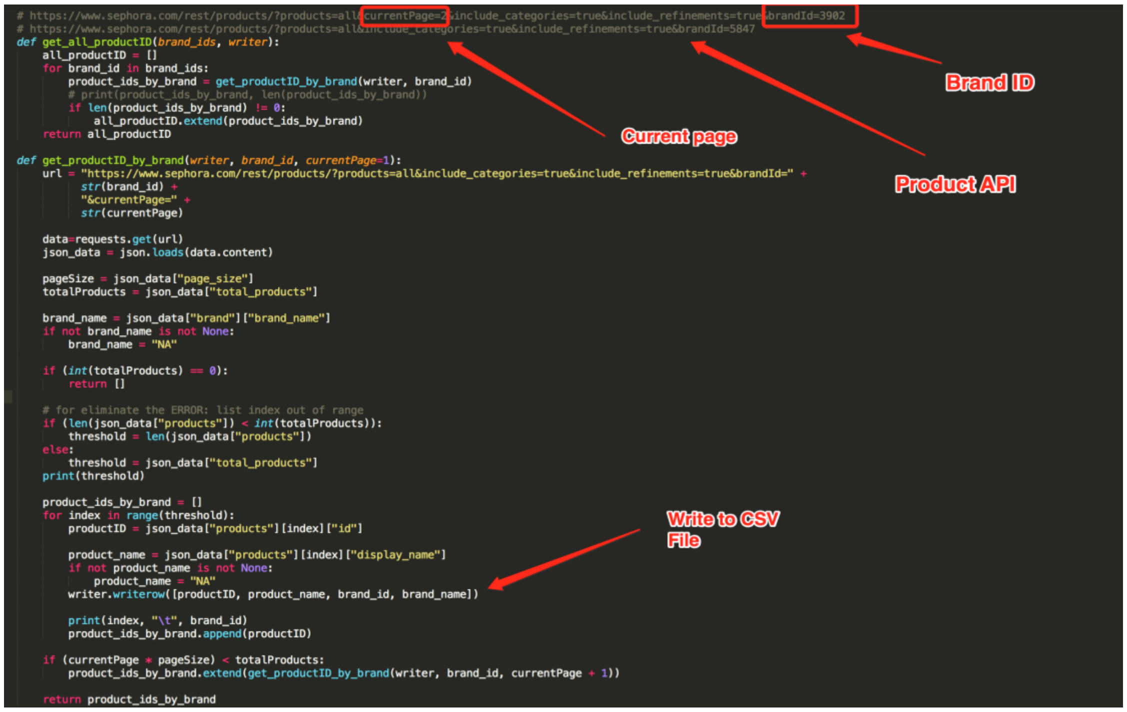
Kun Zhang: Visualizations, Natural language processing, and PPT.

## B: Data scraping detail



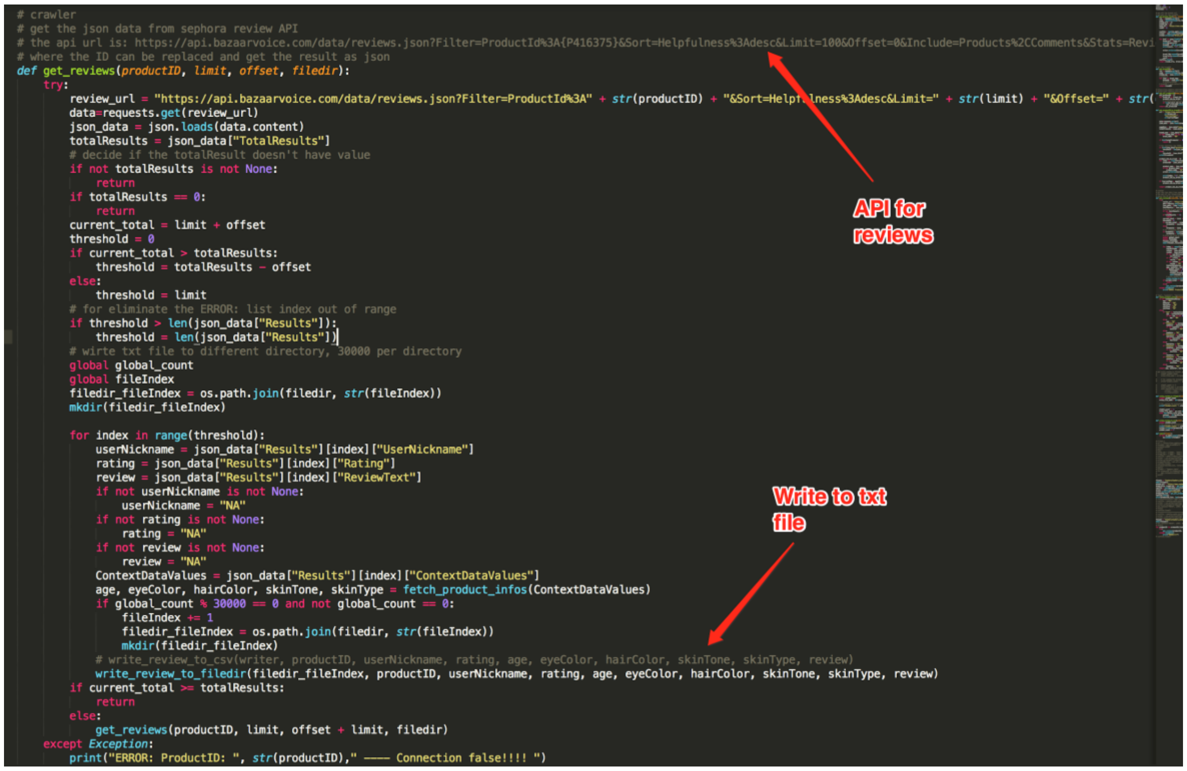
```
# get all the brands url
# and get all the brandID
def get_brands_list(url, filedir):
    res = requests.get(url)
    soup = BeautifulSoup(res.text, "html.parser")
    brand_urls = []
    brand_ids = []
    base_url = "https://www.sephora.com"
    count = 0

    csvfile = open(os.path.join(filedir, "Brands_URL.csv"), "a+")
    writer = csv.writer(csvfile)
    writer.writerow(["Brand_ID", "Brand_URL"])

    for href in soup.findAll("a", {"class": "u-hoverRed u-db u-p1"}):
        brand_url = base_url + href.get('href')
        print(brand_url)
        brandID = get_brandID(brand_url)
        print(count, "\t", brandID)

        writer.writerow([brandID, brand_url])

        brand_ids.append(brandID)
        brand_urls.append(brand_url)
        count += 1
    csvfile.close()
    return brand_urls, brand_ids


# get the brandID
def get_brandID(url):
    res = requests.get(url)
    soup = BeautifulSoup(res.text, "html.parser")
    brand_tag = soup.select("div[certona='BRAND']")
    # print(x)
    for item in brand_tag:
        brandID = item.get("data-brand_id")
        # print(brandID)
    # brandID = soup.select("div seph-search").get('data-brand-id')
    return brandID
```

Figure 18. Code for grabbing brand URL and list of brandID from Web Source Page.



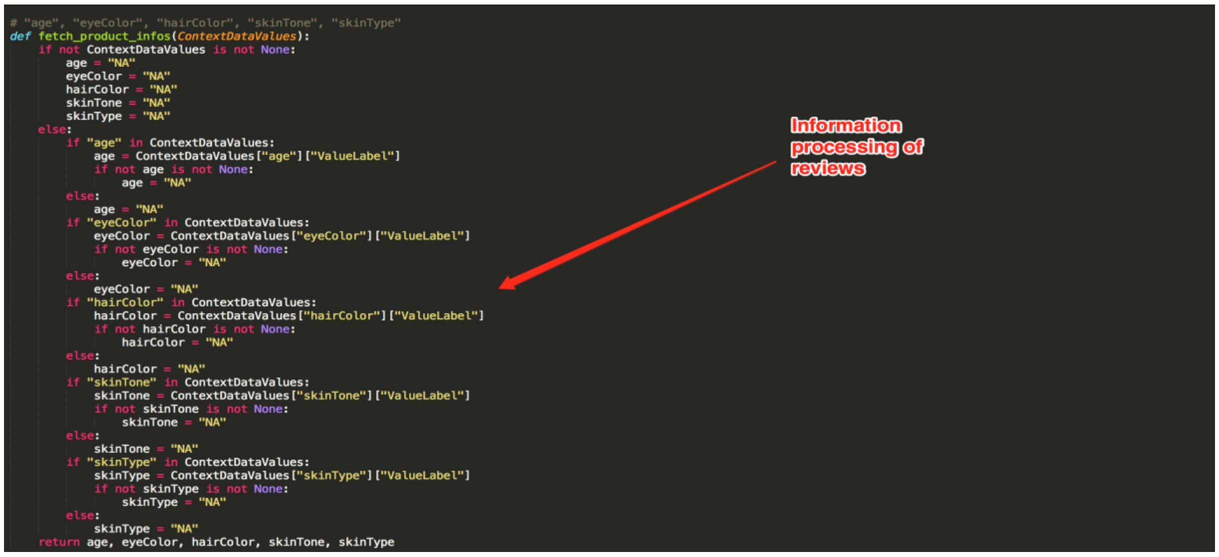
```python
# https://www.sephora.com/rest/products/?products=all&currentPage=2&include_categories=true&include_refinements=true&brandId=3902
# https://www.sephora.com/rest/products/?products=all&include_categories=true&include_refinements=true&brandId=5847
def get_all_productID(brand_ids, writer):
    all_productID = []
    for brand_id in brand_ids:
        product_ids_by_brand = get_productID_by_brand(writer, brand_id)
        # print(product_ids_by_brand, len(product_ids_by_brand))
        if len(product_ids_by_brand) != 0:
            all_productID.extend(product_ids_by_brand)
    return all_productID

def get_productID_by_brand(writer, brand_id, currentPage=1):
    url = "https://www.sephora.com/rest/products/?products=all&include_categories=true&include_refinements=true&brandId=" +
          str(brand_id) +
          "&currentPage=" +
          str(currentPage)

    data=requests.get(url)
    json_data = json.loads(data.content)

    pageSize = json_data["page_size"]
    totalProducts = json_data["total_products"]

    brand_name = json_data["brand"]["brand_name"]
    if not brand_name is not None:
        brand_name = "NA"

    if (int(totalProducts) == 0):
        return []

    # for eliminate the ERROR: list index out of range
    if (len(json_data["products"]) < int(totalProducts)):
        threshold = len(json_data["products"])
    else:
        threshold = json_data["total_products"]
    print(threshold)

    product_ids_by_brand = []
    for index in range(threshold):
        productID = json_data["products"][index]["id"]

        product_name = json_data["products"][index]["display_name"]
        if not product_name is not None:
            product_name = "NA"
        writer.writerow([productID, product_name, brand_id, brand_name])

        print(index, "\t", brand_id)
        product_ids_by_brand.append(productID)

    if (currentPage * pageSize) < totalProducts:
        product_ids_by_brand.extend(get_productID_by_brand(writer, brand_id, currentPage + 1))

    return product_ids_by_brand
```

Figure 19. Code for acquiring all products ID of each brand through productID API.



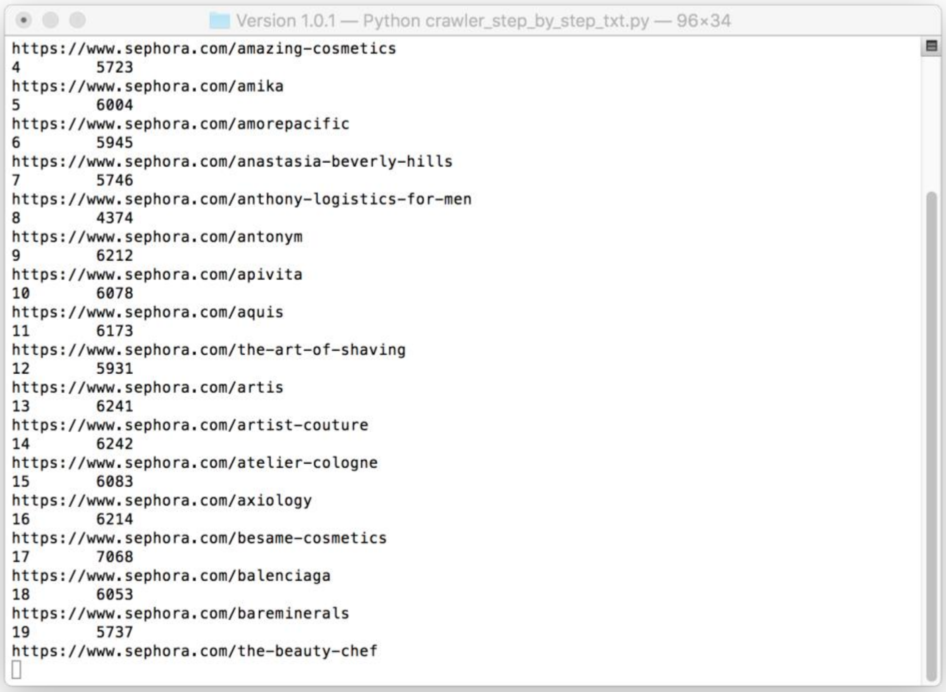
```python
# crawler
# get the json data from sephora review API
# the api url is: https://api.bazaarvoice.com/data/reviews.json?Filter=ProductId%3A{P416375}&Sort=Helpfulness%3Adesc&Limit=100&Offset=0&Include=Products%2CComments&Stats=Revi
# where the ID can be replaced and get the result as json
def get_reviews(productID, limit, offset, filedir):
    try:
        review_url = "https://api.bazaarvoice.com/data/reviews.json?Filter=ProductId%3A" + str(productID) + "&Sort=Helpfulness%3Adesc&Limit=" + str(limit) + "&Offset=" + str(
        data=requests.get(review_url)
        json_data = json.loads(data.content)
        totalResults = json_data["TotalResults"]
        # decide if the totalResult doesn't have value
        if not totalResults is not None:
            return
        if totalResults == 0:
            return
        current_total = limit + offset
        threshold = 0
        if current_total > totalResults:
            threshold = totalResults - offset
        else:
            threshold = limit
        # for eliminate the ERROR: list index out of range
        if threshold > len(json_data["Results"]):
            threshold = len(json_data["Results"])
        # wirte txt file to different directory, 30000 per directory
        global global_count
        global fileIndex
        filedir_fileIndex = os.path.join(filedir, str(fileIndex))
        mkdir(filedir_fileIndex)

        for index in range(threshold):
            userNickname = json_data["Results"][index]["UserNickname"]
            rating = json_data["Results"][index]["Rating"]
            review = json_data["Results"][index]["ReviewText"]
            if not userNickname is not None:
                userNickname = "NA"
            if not rating is not None:
                rating = "NA"
            if not review is not None:
                review = "NA"
            ContextDataValues = json_data["Results"][index]["ContextDataValues"]
            age, eyeColor, hairColor, skinTone, skinType = fetch_product_infos(ContextDataValues)
            if global_count % 30000 == 0 and not global_count == 0:
                fileIndex += 1
                filedir_fileIndex = os.path.join(filedir, str(fileIndex))
                mkdir(filedir_fileIndex)
            # write_review_to_csv(writer, productID, userNickname, rating, age, eyeColor, hairColor, skinTone, skinType, review)
            write_review_to_filedir(filedir_fileIndex, productID, userNickname, rating, age, eyeColor, hairColor, skinTone, skinType, review)
        if current_total >= totalResults:
            return
        else:
            get_reviews(productID, limit, offset + limit, filedir)
    except Exception:
        print("ERROR: ProductID: ", str(productID)," ---- Connection false!!!! ")
```

Figure 20. Code for acquiring reviews for all the product through its ID.



```python
# "age", "eyeColor", "hairColor", "skinTone", "skinType"
def fetch_product_infos(ContextDataValues):
    if not ContextDataValues is not None:
        age = "NA"
        eyeColor = "NA"
        hairColor = "NA"
        skinTone = "NA"
        skinType = "NA"
    else:
        if "age" in ContextDataValues:
            age = ContextDataValues["age"]["ValueLabel"]
            if not age is not None:
                age = "NA"
        else:
            age = "NA"
        if "eyeColor" in ContextDataValues:
            eyeColor = ContextDataValues["eyeColor"]["ValueLabel"]
            if not eyeColor is not None:
                eyeColor = "NA"
        else:
            eyeColor = "NA"
        if "hairColor" in ContextDataValues:
            hairColor = ContextDataValues["hairColor"]["ValueLabel"]
            if not hairColor is not None:
                hairColor = "NA"
        else:
            hairColor = "NA"
        if "skinTone" in ContextDataValues:
            skinTone = ContextDataValues["skinTone"]["ValueLabel"]
            if not skinTone is not None:
                skinTone = "NA"
        else:
            skinTone = "NA"
        if "skinType" in ContextDataValues:
            skinType = ContextDataValues["skinType"]["ValueLabel"]
            if not skinType is not None:
                skinType = "NA"
        else:
            skinType = "NA"
    return age, eyeColor, hairColor, skinTone, skinType
```

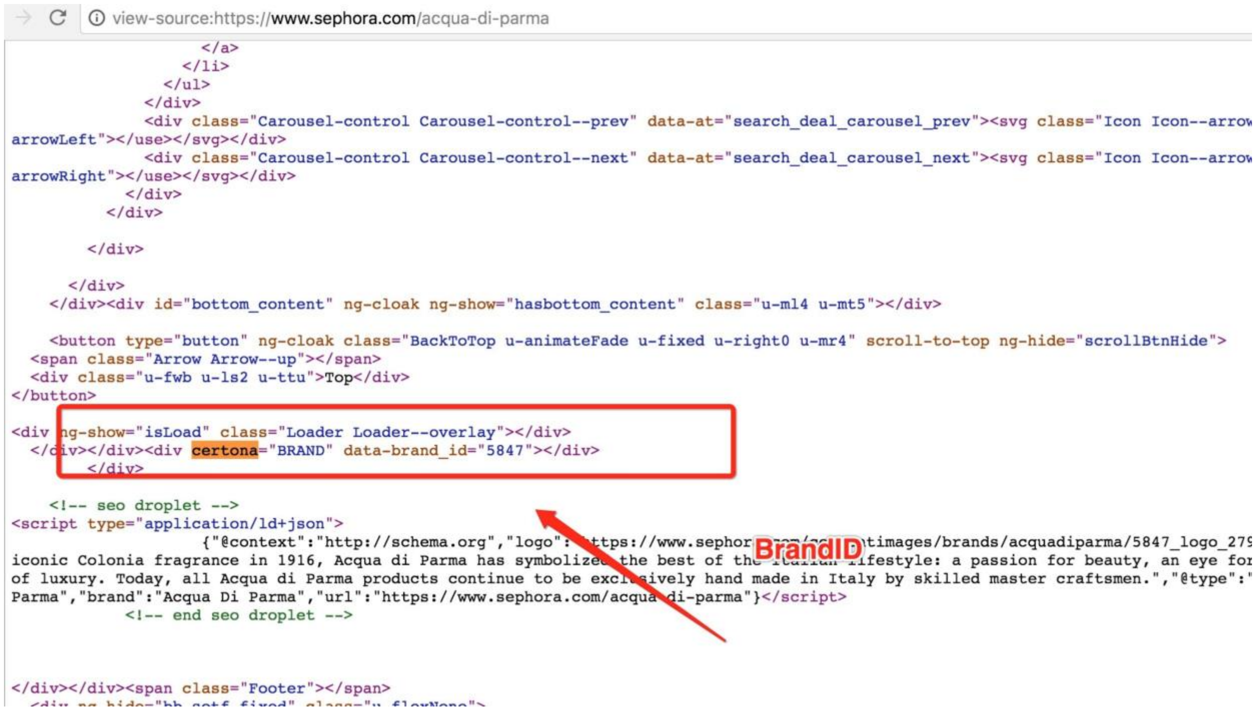
Figure 21. Code for information processing of reviews, placing NONE if the field information does not exist.



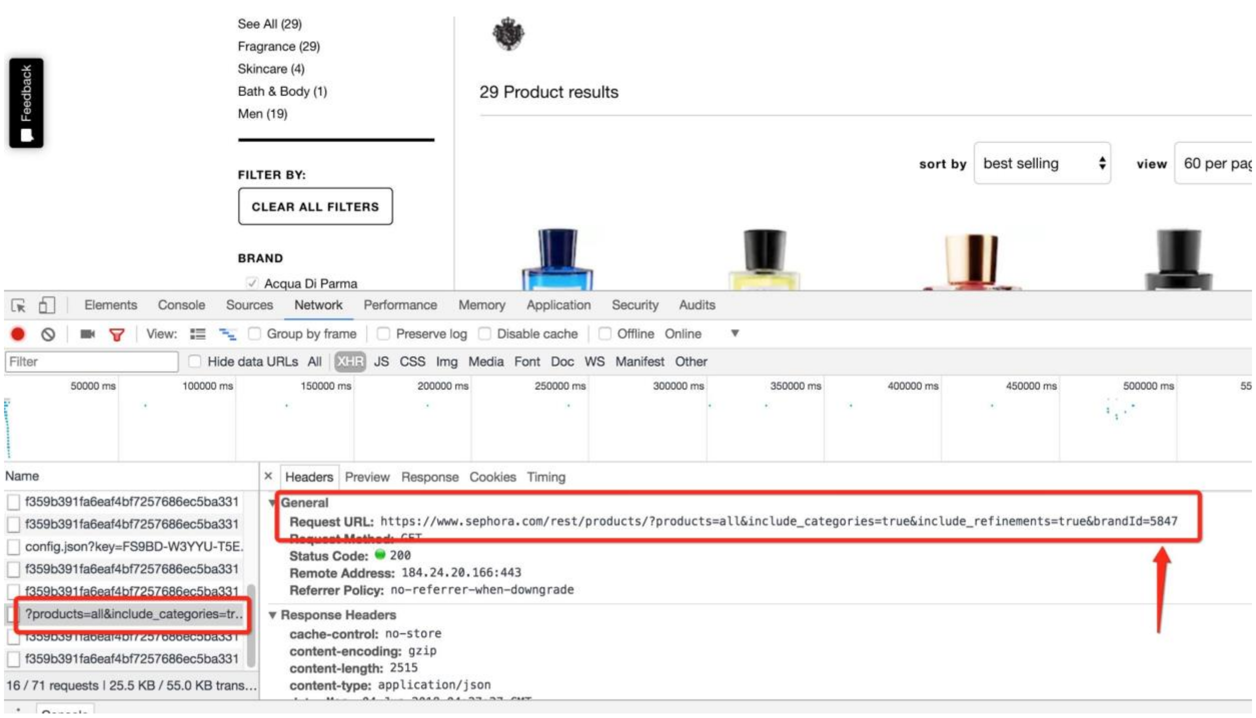
```
https://www.sephora.com/amazing-cosmetics
4       5723
https://www.sephora.com/amika
5       6004
https://www.sephora.com/amorepacific
6       5945
https://www.sephora.com/anastasia-beverly-hills
7       5746
https://www.sephora.com/anthony-logistics-for-men
8       4374
https://www.sephora.com/antonym
9       6212
https://www.sephora.com/apivita
10      6078
https://www.sephora.com/aquis
11      6173
https://www.sephora.com/the-art-of-shaving
12      5931
https://www.sephora.com/artis
13      6241
https://www.sephora.com/artist-couture
14      6242
https://www.sephora.com/atelier-cologne
15      6083
https://www.sephora.com/axiology
16      6214
https://www.sephora.com/besame-cosmetics
17      7068
https://www.sephora.com/balenciaga
18      6053
https://www.sephora.com/bareminerals
19      5737
https://www.sephora.com/the-beauty-chef
```

Figure 22. Run the crawler.

Command prompt: `$ python3 crawler_step_by_step_txt.py`

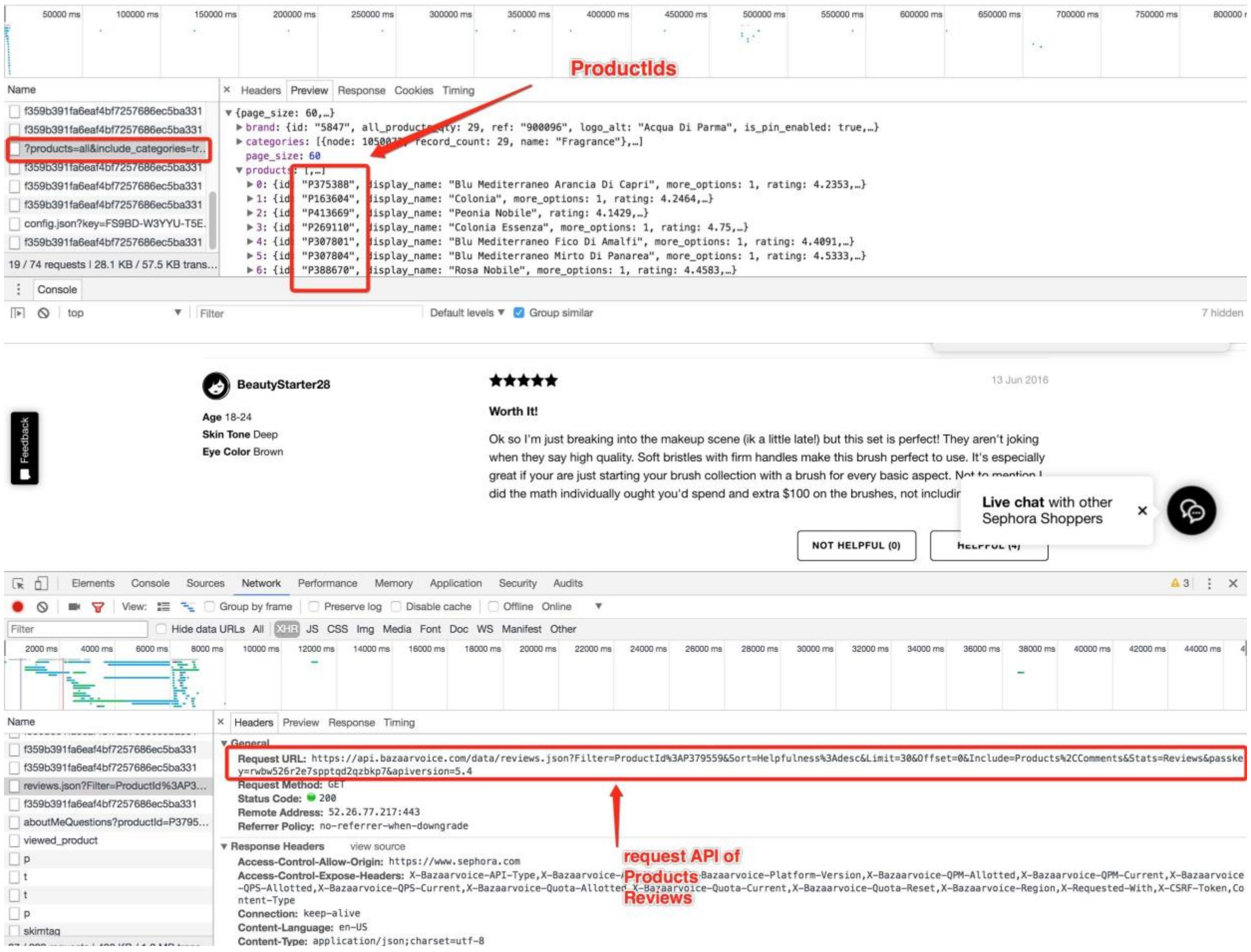

view-source:https://www.sephora.com/acqua-di-parma

```
                    </a>
                  </li>
                </ul>
              </div>
              <div class="Carousel-control Carousel-control--prev" data-at="search_deal_carousel_prev"><svg class="Icon Icon--arrow
arrowLeft"></use></svg></div>
              <div class="Carousel-control Carousel-control--next" data-at="search_deal_carousel_next"><svg class="Icon Icon--arrow
arrowRight"></use></svg></div>
            </div>
          </div>

        </div>

      </div>
    </div><div id="bottom_content" ng-cloak ng-show="hasbottom_content" class="u-ml4 u-mt5"></div>

    <button type="button" ng-cloak class="BackToTop u-animateFade u-fixed u-right0 u-mr4" scroll-to-top ng-hide="scrollBtnHide">
  <span class="Arrow Arrow--up"></span>
  <div class="u-fwb u-ls2 u-ttu">Top</div>
</button>

<div ng-show="isLoad" class="Loader Loader--overlay"></div>
  </div></div><div certona="BRAND" data-brand_id="5847"></div>
        </div>

    <!-- seo droplet -->
<script type="application/ld+json">
                    {"@context":"http://schema.org","logo":"https://www.sephor[illegible]tentimages/brands/acquadiparma/5847_logo_279
iconic Colonia fragrance in 1916, Acqua di Parma has symbolized the best of the Italian lifestyle: a passion for beauty, an eye for
of luxury. Today, all Acqua di Parma products continue to be exclusively hand made in Italy by skilled master craftsmen.","@type":"
Parma","brand":"Acqua Di Parma","url":"https://www.sephora.com/acqua-di-parma"}</script>
            <!-- end seo droplet -->


</div></div><span class="Footer"></span>
```



```
f359b391fa6eaf4bf7257686ec5ba331
f359b391fa6eaf4bf7257686ec5ba331
config.json?key=FS9BD-W3YYU-T5E.
f359b391fa6eaf4bf7257686ec5ba331
f359b391fa6eaf4bf7257686ec5ba331
?products=all&include_categories=tr..
f359b391fa6eaf4bf7257686ec5ba331
f359b391fa6eaf4bf7257686ec5ba331
16 / 71 requests | 25.5 KB / 55.0 KB trans...
```

```
▼ General
    Request URL: https://www.sephora.com/rest/products/?products=all&include_categories=true&include_refinements=true&brandId=5847
    Request Method: GET
    Status Code: ● 200
    Remote Address: 184.24.20.166:443
    Referrer Policy: no-referrer-when-downgrade
▼ Response Headers
    cache-control: no-store
    content-encoding: gzip
    content-length: 2515
    content-type: application/json
```



```
Name
f359b391fa6eaf4bf7257686ec5ba331
f359b391fa6eaf4bf7257686ec5ba331
?products=all&include_categories=tr..
f359b391fa6eaf4bf7257686ec5ba331
f359b391fa6eaf4bf7257686ec5ba331
f359b391fa6eaf4bf7257686ec5ba331
config.json?key=FS9BD-W3YYU-T5E.
f359b391fa6eaf4bf7257686ec5ba331
19 / 74 requests | 28.1 KB / 57.5 KB trans...
```

```
▼{page_size: 60,…}
 ▶brand: {id: "5847", all_product_qty: 29, ref: "900096", logo_alt: "Acqua Di Parma", is_pin_enabled: true,…}
 ▶categories: [{node: 1050072, record_count: 29, name: "Fragrance"},…]
  page_size: 60
 ▼products: [,…]
  ▶0: {id: "P375388", display_name: "Blu Mediterraneo Arancia Di Capri", more_options: 1, rating: 4.2353,…}
  ▶1: {id: "P163604", display_name: "Colonia", more_options: 1, rating: 4.2464,…}
  ▶2: {id: "P413669", display_name: "Peonia Nobile", rating: 4.1429,…}
  ▶3: {id: "P269110", display_name: "Colonia Essenza", more_options: 1, rating: 4.75,…}
  ▶4: {id: "P307801", display_name: "Blu Mediterraneo Fico Di Amalfi", more_options: 1, rating: 4.4091,…}
  ▶5: {id: "P307804", display_name: "Blu Mediterraneo Mirto Di Panarea", more_options: 1, rating: 4.5333,…}
  ▶6: {id: "P388670", display_name: "Rosa Nobile", more_options: 1, rating: 4.4583,…}
```

BeautyStarter28 ★★★★★ 13 Jun 2016

Age 18-24
Skin Tone Deep
Eye Color Brown

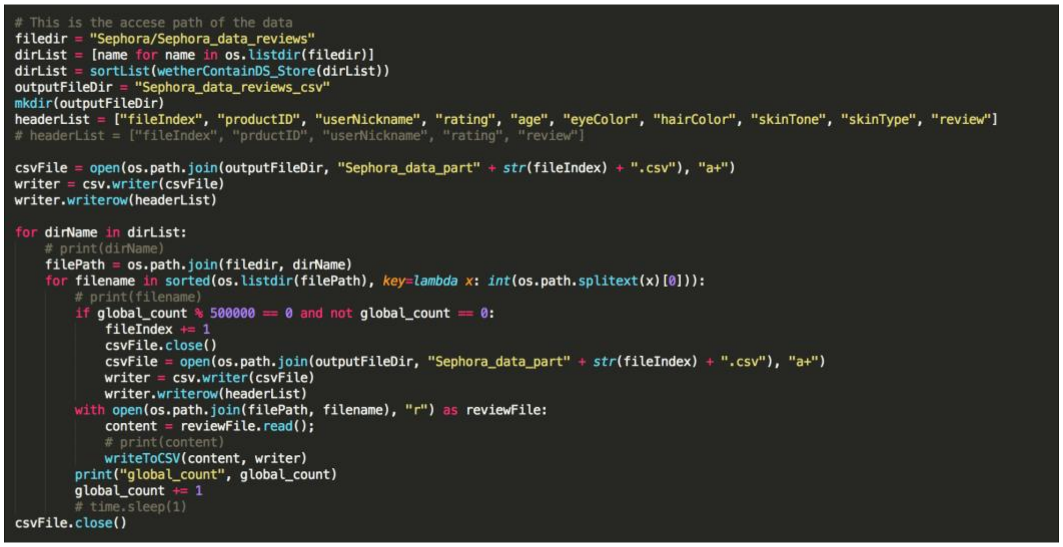
Worth It!

Ok so I'm just breaking into the makeup scene (ik a little late!) but this set is perfect! They aren't joking when they say high quality. Soft bristles with firm handles make this brush perfect to use. It's especially great if your are just starting your brush collection with a brush for every basic aspect. Not to mention I did the math individually ought you'd spend and extra $100 on the brushes, not includir

```
Name
f359b391fa6eaf4bf7257686ec5ba331
f359b391fa6eaf4bf7257686ec5ba331
reviews.json?Filter=ProductId%3AP3...
f359b391fa6eaf4bf7257686ec5ba331
aboutMeQuestions?productId=P3795...
viewed_product
p
t
t
p
skimtag
```

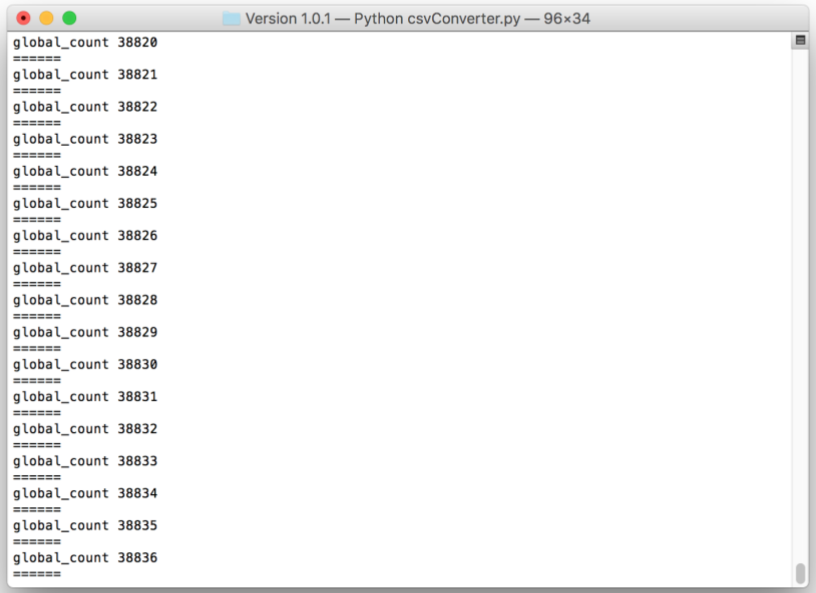
```
▼General
 Request URL: https://api.bazaarvoice.com/data/reviews.json?Filter=ProductId%3AP379559&Sort=Helpfulness%3Adesc&Limit=30&Offset=0&Include=Products%2CComments&Stats=Reviews&passkey=rwbw526r2e7spptqd2qzbkp7&apiversion=5.4
 Request Method: GET
 Status Code: 200
 Remote Address: 52.26.77.217:443
 Referrer Policy: no-referrer-when-downgrade
▼Response Headers    view source
 Access-Control-Allow-Origin: https://www.sephora.com
 Access-Control-Expose-Headers: X-Bazaarvoice-API-Type,X-Bazaarvoice-…Bazaarvoice-Platform-Version,X-Bazaarvoice-QPM-Allotted,X-Bazaarvoice-QPM-Current,X-Bazaarvoice-QPS-Allotted,X-Bazaarvoice-QPS-Current,X-Bazaarvoice-Quota-Allotted,X-Bazaarvoice-Quota-Current,X-Bazaarvoice-Quota-Reset,X-Bazaarvoice-Region,X-Requested-With,X-CSRF-Token,Content-Type
 Connection: keep-alive
 Content-Language: en-US
 Content-Type: application/json;charset=utf-8
```

## C: Transforming TXT into CSV details



```python
# This is the accese path of the data
filedir = "Sephora/Sephora_data_reviews"
dirList = [name for name in os.listdir(filedir)]
dirList = sortList(wetherContainDS_Store(dirList))
outputFileDir = "Sephora_data_reviews_csv"
mkdir(outputFileDir)
headerList = ["fileIndex", "productID", "userNickname", "rating", "age", "eyeColor", "hairColor", "skinTone", "skinType", "review"]
# headerList = ["fileIndex", "prductID", "userNickname", "rating", "review"]

csvFile = open(os.path.join(outputFileDir, "Sephora_data_part" + str(fileIndex) + ".csv"), "a+")
writer = csv.writer(csvFile)
writer.writerow(headerList)

for dirName in dirList:
    # print(dirName)
    filePath = os.path.join(filedir, dirName)
    for filename in sorted(os.listdir(filePath), key=lambda x: int(os.path.splitext(x)[0])):
        # print(filename)
        if global_count % 500000 == 0 and not global_count == 0:
            fileIndex += 1
            csvFile.close()
            csvFile = open(os.path.join(outputFileDir, "Sephora_data_part" + str(fileIndex) + ".csv"), "a+")
            writer = csv.writer(csvFile)
            writer.writerow(headerList)
        with open(os.path.join(filePath, filename), "r") as reviewFile:
            content = reviewFile.read();
            # print(content)
            writeToCSV(content, writer)
        print("global_count", global_count)
        global_count += 1
        # time.sleep(1)
csvFile.close()
```

Figure 23. Reading all the reviews information from text file, and writing to CSV file.

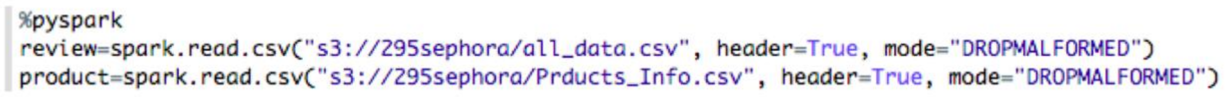


```
global_count 38820
======
global_count 38821
======
global_count 38822
======
global_count 38823
======
global_count 38824
======
global_count 38825
======
global_count 38826
======
global_count 38827
======
global_count 38828
======
global_count 38829
======
global_count 38830
======
global_count 38831
======
global_count 38832
======
global_count 38833
======
global_count 38834
======
global_count 38835
======
global_count 38836
======
```

Figure 24. Run the csvConverter, which is used to read reviews from all the text file and store into CSV files.

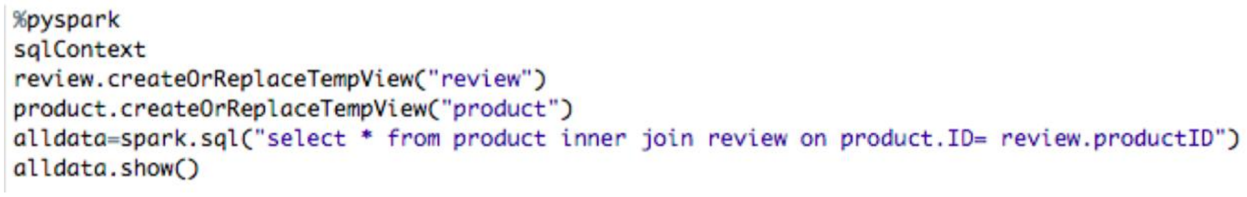
Command prompt: $ `python3 csvConverter.py`

## D: Processing data through Spark



```
%pyspark
review=spark.read.csv("s3://295sephora/all_data.csv", header=True, mode="DROPMALFORMED")
product=spark.read.csv("s3://295sephora/Prducts_Info.csv", header=True, mode="DROPMALFORMED")
```

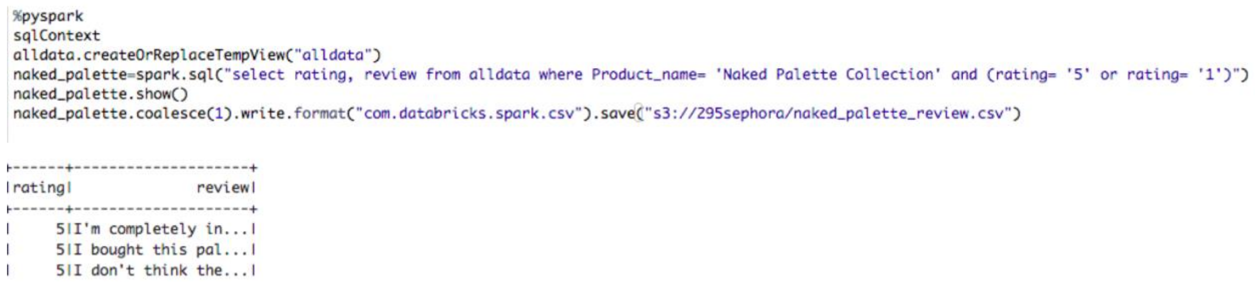
Figure.25 Spark code to create RDD



```
%pyspark
sqlContext
review.createOrReplaceTempView("review")
product.createOrReplaceTempView("product")
alldata=spark.sql("select * from product inner join review on product.ID= review.productID")
alldata.show()
```

Figure.26 Spark code to create temp table

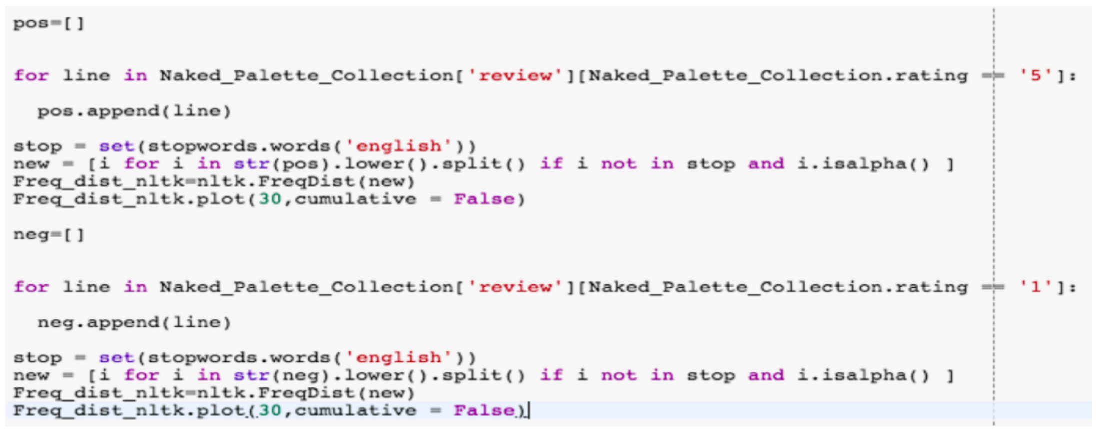
## E: Word Frequency Analysis



```
%pyspark
sqlContext
alldata.createOrReplaceTempView("alldata")
naked_palette=spark.sql("select rating, review from alldata where Product_name= 'Naked Palette Collection' and (rating= '5' or rating= '1')")
naked_palette.show()
naked_palette.coalesce(1).write.format("com.databricks.spark.csv").save("s3://295sephora/naked_palette_review.csv")
```

```
+------+--------------------+
|rating|              review|
+------+--------------------+
|     5|I'm completely in...|
|     5|I bought this pal...|
|     5|I don't think the...|
```

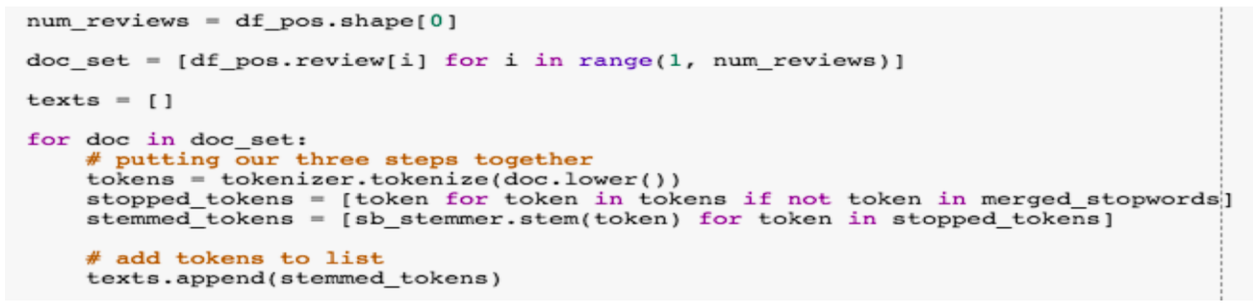
Figure.27 code to filter and save qualified data



```
pos=[]

for line in Naked_Palette_Collection['review'][Naked_Palette_Collection.rating == '5']:

  pos.append(line)

stop = set(stopwords.words('english'))
new = [i for i in str(pos).lower().split() if i not in stop and i.isalpha() ]
Freq_dist_nltk=nltk.FreqDist(new)
Freq_dist_nltk.plot(30,cumulative = False)

neg=[]

for line in Naked_Palette_Collection['review'][Naked_Palette_Collection.rating == '1']:

  neg.append(line)

stop = set(stopwords.words('english'))
new = [i for i in str(neg).lower().split() if i not in stop and i.isalpha() ]
Freq_dist_nltk=nltk.FreqDist(new)
Freq_dist_nltk.plot(30,cumulative = False)
```

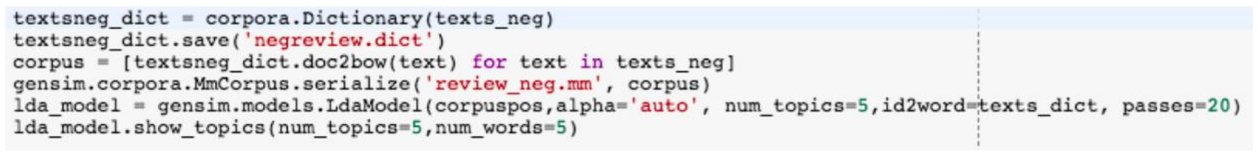
Figure.28 code for word frequency analysis

## F: LDA model



```
num_reviews = df_pos.shape[0]

doc_set = [df_pos.review[i] for i in range(1, num_reviews)]

texts = []

for doc in doc_set:
    # putting our three steps together
    tokens = tokenizer.tokenize(doc.lower())
    stopped_tokens = [token for token in tokens if not token in merged_stopwords]
    stemmed_tokens = [sb_stemmer.stem(token) for token in stopped_tokens]

    # add tokens to list
    texts.append(stemmed_tokens)
```

Figure.29 code for preprocessing natural language data



```
textsneg_dict = corpora.Dictionary(texts_neg)
textsneg_dict.save('negreview.dict')
corpus = [textsneg_dict.doc2bow(text) for text in texts_neg]
gensim.corpora.MmCorpus.serialize('review_neg.mm', corpus)
lda_model = gensim.models.LdaModel(corpuspos,alpha='auto', num_topics=5,id2word=texts_dict, passes=20)
lda_model.show_topics(num_topics=5,num_words=5)
```

Figure.30 code for LDA model