Assignment 1 Report

Abstract

In the modern society, more and more people are paying attention to their facial beauty. Our team project is about accessing makeup product data from different recourses, build the database schema and create the data table, which shows the detailed information of the Loreal makeup products.

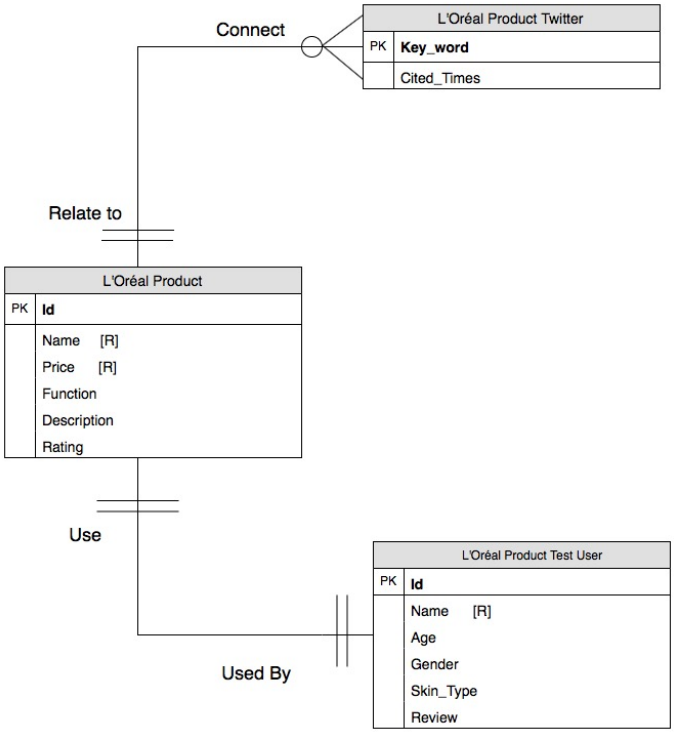
Data Sources

We downloaded the raw csv data about the information of Loreal makeup products test users in www.wenku.baidu.com. We obtained the Twitter related information of Loreal makeup products from Twitter API and we used web scraper to get data from Loreal official website.

Conceptual Schema Explanation

The data from three sources are related to each other. Every test user only uses one product and each product only be used by one user. The results of the tweets are related to only one product. One Loreal product can connect with zero or more tweets.

Object model diagram



Report of our code

- **Web Scraper**

```
In [95]: from requests import get

In [96]: url = "https://www.lorealparisusa.com/products/makeup/shop-all-products.aspx"

In [97]: response = get(url)
         response
Out[97]: <Response [200]>
```

1. Access data from https://www.lorealparisusa.com/products/makeup/shop-all-products.aspx and the status shows obtaining data successfully

```
In [98]: from bs4 import BeautifulSoup as bs
         html_soup = bs(response.content,'html.parser')
         type(html_soup)
         print(html_soup.prettify())
```

2. Import the BeautifulSoup to parse the website and check the whole html

```
In [99]: product_container =html_soup.find_all('div',class_ ="subcat-product-box")
         print(len(product_container))
         product_container
```

3. Obtain the product container for each item and the check how many products show in the website.

```
In [102]: container = product_container[0]


          cosmetic_names = []
          cosmetic_price = []
          cosmetic_function = []
          cosmetic_description = []
          cosmetic_rating = []
```

4. Get one specific container for getting data and create different empty lists to store data we will obtain from the website.

In [103]:
```
for container in product_container:

    name = container.h3.text
    cosmetic_names.append(name)

for container in product_container:
 price = container.find("span",class_ = "product-container__actions--price").text
 cosmetic_price.append(price)

for container in product_container:

    function = container.h4.text
    cosmetic_function.append(function)

for container in product_container:

    description = container.p.text
    cosmetic_description.append(description)

for container in product_container:
 rating = container.find("ul",class_ = "rating-selected").text
 cosmetic_rating.append(rating)
```

5. Get the text of the name, price, cosmetic function, cosmetic description and the rating from each container and add it to their list.

In [104]:
```
import pandas as pd
```

In [105]:
```
test_df = pd.DataFrame({
        "L'Oreal Paris_Product_names": cosmetic_function,
        "L'Oreal Paris_Product_price": cosmetic_price,
        "L'Oreal Paris_Product_function": cosmetic_names,
        "L'Oreal Paris_Product_description": cosmetic_description,
        "L'Oreal Paris_Product_rating": cosmetic_rating,
 })

test_df
```

6. Import pandas to transform the lists we get from the website to the table.

In [106]:
```
test_df.to_csv("L'Oreal Paris_Product.csv")
```

7. Import out the chart as csv file.

- **Web API**

In [24]:
```
import twitter

CONSUMER_KEY = 'NDSPjwGO2oX0Z08h98ro3eSJB'
CONSUMER_SECRET = 'hvAqEQBHIZmPTEmrgZDVu7fgqm4jTwOEUwlFje00GYunFEVfxt'
OAUTH_TOKEN = '1086338263206514689-h4JGnBWAIRXUAKTBLh3E9jAD5vfUNh'
OAUTH_TOKEN_SECRET = 'scBdFvlrjQq0SHFQiG0euTDfScrfaFlTzzglfEwpgF4fW'

auth = twitter.oauth.OAuth(OAUTH_TOKEN, OAUTH_TOKEN_SECRET,
                           CONSUMER_KEY, CONSUMER_SECRET)

twitter_api = twitter.Twitter(auth=auth)



print(twitter_api)
```

```
<twitter.api.Twitter object at 0x10f6a6f28>
```

1. Access data from Twitter API, and provide cosumer_key, consumer_secret, oauth_token, oauth_token_secret for authorization. And the output shows access successfully.

In [25]:
```
import re

def count_occurance(word, json_to_search):
    a = re.split(r'\W', str(json_to_search).lower())
    return a.count(word.lower())
```

2. Create a method which can count the occurrences of the keyword (which is the product name)

In [26]:
```
import json
from urllib.parse import unquote
```

In [27]:
```
dict_tags = ['Colour Riche Nude', 'Rouge Signature', 'True Match Lumi', 'Unlimited Waterproof', 'Infallible 24 HR',
             'Infallible Full', 'Unlimited Washable', 'UnbelievaBrow', 'Colour Riche Stain', 'Colour Riche Lipstick',
             'Voluminous', 'Infallible Gloss', 'Infallible Highlighter', 'Infallible Eye Shadow ', 'Voluminous Mascara',
             'Paradise Eyeshadow', 'Voluminous Eyeliner', 'Infallible Highlighter Sticks', 'Paradise Blush', 'Infallible
```

In [28]:
```
d = {}
for word in dict_tags:
    search_results = twitter_api.search.tweets(q=word, count=500)
    statuses = search_results['statuses']
    d[word] = statuses
```

3. Create a dictionary (dict_tags) to store all the keywords, then use for loop to repeat the search related tweets of keyword. And then put the keyword and its results in another dictionary named d.

In [30]:
```
import csv
with open('output.csv', 'w', newline='') as csvfile:
    fieldnames = ['Twitter_Keyword', 'Twitter_Search_Result_Counts']
    writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
    writer.writeheader()

# Counting the number of occurences of the word in the json string and storing it
    for word in dict_tags:
        count = count_occurance(word, d[word])
        #Write the count and the word in csv here.
        writer.writerow({'Twitter_Keyword': word,'Twitter_Search_Result_Counts': str(count) })
        print(word,":", str(count))
```

```
Colour Riche Nude : 0
Rouge Signature : 0
True Match Lumi : 0
Unlimited Waterproof : 0
Infallible 24 HR : 0
Infallible Full : 0
Unlimited Washable : 0
UnbelievaBrow : 13
Colour Riche Stain : 0
Colour Riche Lipstick : 0
Voluminous : 130
Infallible Gloss : 0
Infallible Highlighter : 0
Infallible Eye Shadow  : 0
Voluminous Mascara : 0
Infallible Foundation Sticks : 0
Paradise Eyeshadow : 0
Voluminous Eyeliner : 0
Infallible Highlighter Sticks : 0
Paradise Blush : 0
Infallible Blush Sticks  : 0
```

4. Use the method at the beginning of the code to get the occurrences of the words in the tweets, and import the csv, so we can store the results in csv file.

- **Data Cleaning**

In [185]:
```
import numpy as np
import pandas as pd
# loandata = pd.DataFrame('products.csv')
df1 = pd.read_csv("L'Oreal Paris_Product.csv",encoding='latin-1');
df2 = pd.read_csv("Test_User_L'Oreal Paris_Product .csv",encoding='latin-1');
df3 = pd.read_csv("Twitter_API_output.csv",encoding='latin-1');
# df = pd.merge(df1, df2,  how= 'left' )
data = pd.concat([df1,df2,df3],axis=1)
```

In [186]:
```
data
```

1. Read all the csv files and combine them in the same table.

In [187]:
```
data = data.fillna('unavailable')
```

2. Fill all the empty data as “unavailable”.

```
In [188]: data["L'Oreal Paris_Product_price"].apply(lambda x: x. isdigit ())

Out[188]: 0     False
          1     False
          2     False
          3     False
          4     False
          5     False
          6     False
          7     False
          8     False
          9     False
          10    False
          11    False
          12    False
          13    False
          点击滚动输出;双击隐藏
          15    False
          16    False
          17    False
          18    False
          19    False
          20    False
          Name: L'Oreal Paris_Product_price, dtype: bool

In [ ]:

In [189]: data['Test_User_Gender']= data['Test_User_Gender'].map(str.strip)
```

3. Check whether the price data is only digit and strip the blank of the user gender.

```
In [191]: data = data.drop_duplicates()

In [192]: data.describe().astype(np.int64).T
```

Out[192]:

	count	mean	std	min	25%	50%	75%	max
Unnamed: 0	21	10	6	0	5	10	15	20
Unnamed: 0	21	10	6	0	5	10	15	20
Test_User_Age	21	23	2	19	22	23	25	27
Twitter_Search_Result_Counts	21	6	28	0	0	0	0	130

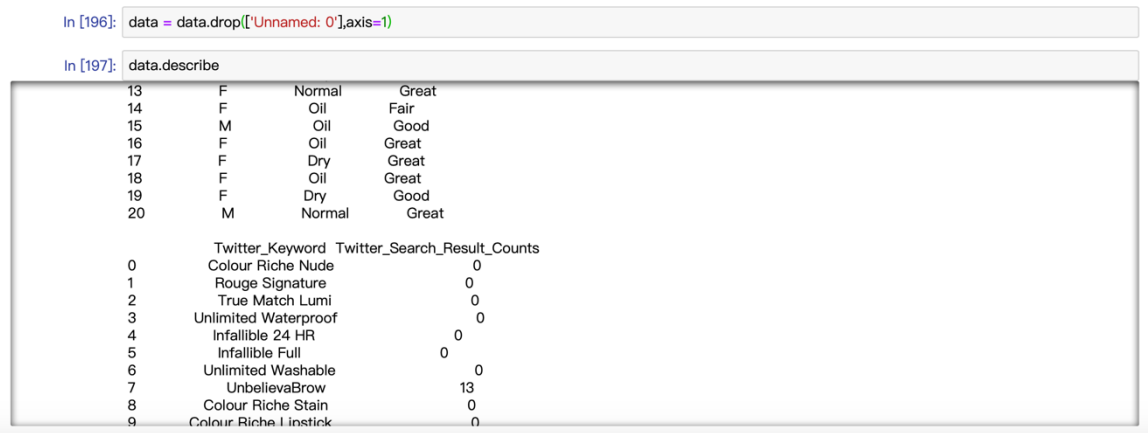
4. Drop the duplicated data and check the type of our data.

```
In [193]: # loandata.dtypes
          data = data.replace([30,18],data['Test_User_Age'].mean())
          data = data.replace("\n\n\n\n\n\n","unavailable")

In [194]: data['Test_User_Name']=data['Test_User_Name'].map(str.title)

In [195]: data
```

5. Replace the abnormal age under 18 and over 30 with the mean of the age. Since the rating of the products are not available now, so we replace with the word “unavailable ”. And we check whether the first letter of test user name is big.



6. Drop the useless line and check all the data one more time.

Conclusion and Inference

Most users consider Loreal makeup products are in good functions according to the data we collected. And the product named Voluminous considered the most popular since the product was searched the most times.

Citations and Reference

https://github.com/nikbearbrown/INFO_6210/blob/master/Week_2/NBB_IMDB_Web_Scraper.ipynb
https://github.com/nikbearbrown/INFO_6210/blob/master/Week_2/NBB_Social_Web_Twitter.ipynb

Text license and Percentage contribution in assignment