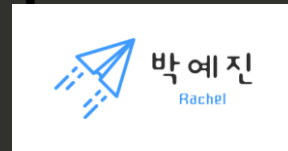


코딩 프로젝트 포트폴리오



Python, Jupiter Notebook

건강에 대한 관심 高

2020년 코로나19 팬데믹 상황에서, 소비자들은 면역력 증진에 대한 관심이 높아졌다. 면역력 증진에 효과적인 홍삼이 주목받으면서, 시중에 유통 중인 많은 홍삼제품의 유효성분, 가격을 따져보며 구매하는 현명한 소비자들이 늘고 있다.

#1. 홍삼 프로젝트

가격 비교 사이트에 등록된 홍삼 데이터를 토대로 홍삼 제품의 가성비 분석

왜 이 프로젝트를 시작했나?

건강식품 관심 증대
코로나19 상황에서, 면역
력을 강화하려는 소비자들
의 수요가 증가함

독점적 홍삼시장구조
→ **높은 구매장벽**
홍삼 No.1 브랜드는 고
가의 가격대를 형성하
고 있어 소비자는 가격
에 대한 부담으로 홍삼
구매를 주저함

군소 홍삼제품 난립

다양한 군소 브랜드들이
치열한 경쟁을 벌이나
No.1 브랜드를 견제할
주목받는 브랜드가 부재함

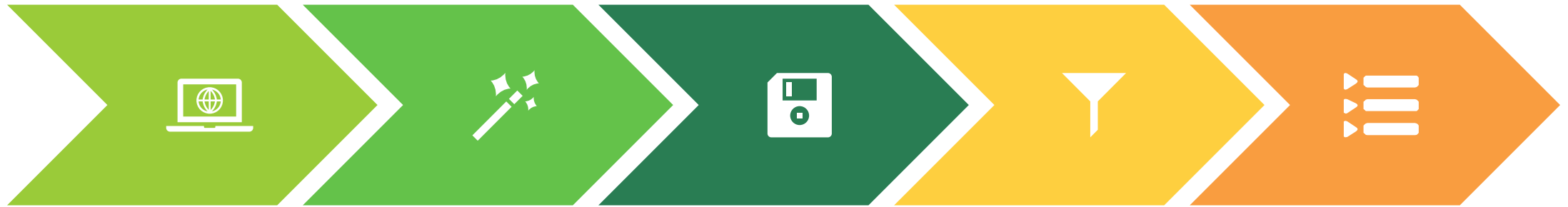
홍삼에 대한 소비자들의 지식 부족

홍삼이 몸에 좋은 것은 알고 있으
나, 구체적으로 어떤 제품이
왜 좋은지는 모르는 소비자 多

가성비 좋은 홍삼 수요 증가

브랜드만 보고 구매하기 보다
는 가성비가 좋은 홍삼 제품
을 선택하는 똑똑한 소비자들
이 늘고 있음

프로젝트 Flow



다나와 웹크롤링

BeautifulSoup을 활용하여,
다나와 사이트의 홍삼자료에
접근하여 HTML 데이터를
수집

URL 자동생성 함수

여러 페이지에 있는 자료를
자동으로 수집하기 위해 URL
주소를 분석하고, URL을
생성하는 함수를 구현함

데이터 추출 및 저장

HTML데이터에서 분석에
필요한 데이터만 1차로
추출하여 엑셀로 저장

데이터 정제

한 컬럼에 포함된 여러
데이터를 분리하고, 분석할
수 있는 상태로 데이터를
정제함

데이터 가공 및 분석

홍삼 유효성분인
진세노사이드 1mg당 제품
가격이 저렴한 순으로
홍삼제품을 정렬함

활용

사용 언어

Python



사용 환경

Jupyter Notebook

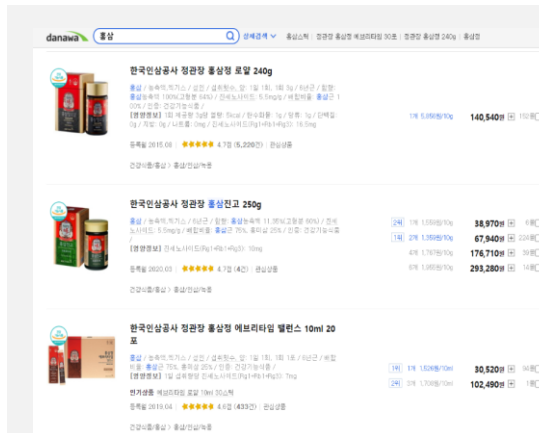


활용 라이브러리

Selenium
BeautifulSoup
Time
pandas



1. 다나와 웹크롤링 & 데이터 추출 및 저장



사이트 구조 분석

- 필요한 정보가 포함된 태그, 클래스명 등을 파악함

```
In [42]: def get_items(net_items):
product_data = []

for item in net_items:
    try:
        title = item.select('p.prod_name > a')[0].text.strip()
    except:
        title = ''
    try:
        spec_list = item.select('div.spec_list')[0].text.strip()
    except:
        spec_list = ''
    try:
        price = item.select('div.prod_price_list > ul > li.rank_one > p > a > strong')[0].text.strip().replace(',', '')
    except:
        price = 0
    try:
        unit_price = item.select('div.prod_price_list > ul > li.rank_one > p > a > span.new_price_text > em.lowest')[0].text.strip()
    except:
        unit_price = 0

    product_data.append([title, spec_list, price, unit_price])

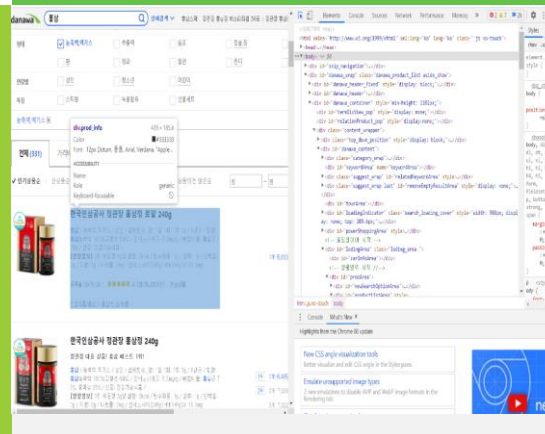
return product_data
```

데이터 추출 및 저장

- 제품명, 스펙리스트, 가격, 단가에 대한 정보를 리스트로 저장

다나와 사이트 접속

- 검색어 : 홍삼
- 형태 필터: 농축액, 엑기스
- 40건씩 총 9페이지



함수 작성

- 제품명, 스펙리스트, 가격, 단가로 구분해 저장하는 함수 작성
- URL 자동 생성 함수로 1~9페이지까지 추출

```
In [48]: from selenium import webdriver
import time
from bs4 import BeautifulSoup
from tqdm.notebook import tqdm

driver = webdriver.Chrome('C:/Users/jeann/Documents/playwithdata/chromedriver.exe')

driver.implicitly_wait(3)

keyword = '홍삼'
total_page = 9
prod_data_total = []

for page in tqdm(range(1, total_page + 1)):
    url = get_search_page_url(keyword, page)
    driver.get(url)
    time.sleep(5)

    html = driver.page_source
    soup = BeautifulSoup(html, 'html.parser')

    net_items = soup.select("[id='productItem']")
    item_list = get_items(net_items)

    prod_data_total += item_list
```


2. 데이터 정제, 가공

1

```
# 10g당 단가를 10g당 진세노사이드 함량으로 나눠서 진세노사이드 1mg 당 단가를 추출하기 위한 전처리 (10g당 단가/10g당 진세노사이드 함량)
unit_price = []

for cost in data['단가']:
    if cost != 0:
        price = int(cost.split('원')[0])
    elif cost == 0:
        price = 0
    else:
        price = 0
    unit_price.append(price)

for detail in data['스펙목록']:
    spec_list = detail.split('/')

    for spec in spec_list:
        if '진세노사이드' in spec:
            ginseno = spec
            ginseno_amount = ginseno.split(':')[1]
            ginseno_per1g = float(ginseno_amount.split('mg')[0])
            ginseno_per10g = ginseno_per1g * 10

            ginseno_portion.append(ginseno_per10g)

print("10g당 진세노함량: ", len(ginseno_portion), ginseno_portion[0:10])
print("10g당 가격: ", len(unit_price), unit_price[0:10])

10g당 진세노함량: 335 [55.0, 55.0, 55.0, 41.0, 55.0, 55.0, 55.0, 40.0, 60.0, 55.0]
10g당 가격: 335 [5693, 0, 6640, 8400, 1292, 7358, 5234, 281, 0, 3034]
```

- 단가 정보에서 정수부분만 분리
- 스펙목록 내 다양한 정보를 구분기호(/)로 분리하고, 1g당 진세노사이드 함량(mg)을 정수부분만 추출하고 10을 곱함
- 10g 당 진세노사이드 함량과 10g 당 가격 정보 추출

2

```
# 홍삼농축액 함량만 따로 빼서 새로운 컬럼으로 구성하고, 이 중에서 100%인 행만 추출하고자 함
concentrate = []
```

```
for spec_data in data['스펙목록']:
    spec_list = spec_data.split('/')

    percentile = None

    for spec in spec_list:
        if '홍삼농축액' in spec:
            percent = spec.split('홍삼농축액')[1]
            percentile = percent.split('%')[0].strip()

    concentrate.append(percentile)
```

- 홍삼농축액 함량만 추출해 새로운 컬럼으로 생성함
(추후 홍삼농축액 100%인 제품만 비교하기 위한 사전 작업)

3

```
# 10g당 단가를 10g당 진세노사이드 함량으로 나눠서 진세노사이드 1mg 당 단가 추출작업 (10g당 단가/10g당 진세노사이드 함량)
final_unit_price = []
```

```
for i in range(len(pd_data)):
    unit_cost = pd_data['10g당 가격'][i]/pd_data['10g당 진세노함량'][i]

    final_unit_price.append(unit_cost)

print("진세노 1mg당 가격: ", len(final_unit_price), final_unit_price[0:10])
```

```
진세노 1mg당 가격: 335 [103.50909090909092, 0.0, 120.72727272727273, 204.8780487804878, 23.490909090909092, 133.78181818181818, 95.16363636363636, 7.025, 0.0, 55.163636363636364]
```

- 진세노사이드 1mg당 가격 계산
: 10g 당 가격 / 10g당 진세노사이드 함량

4

```
# 홍삼농축액 함량 100%인 제품만 비교하기 위해 홍삼농축액 함량 100%인 제품만 필터링을 걸고, 이를 오름차순으로 정리함
condition = pd_data['홍삼농축액 함량'].isin(['100'])
condition2 = (pd_data['10g당 가격']!=0)
data1 = pd_data[condition]
data2 = data1[condition2 == False]
final_list = data2.sort_values(['진세노 1mg당 가격'], ascending=True)
final_list.head(10)
```

- 조건 : 홍삼 농축액이 100%인 제품 리스트만 필터링
- 진세노사이드 1mg당 가격을 오름차순으로 정리

3. 데이터 분석 및 검증

- 총 78개의 제품 리스트

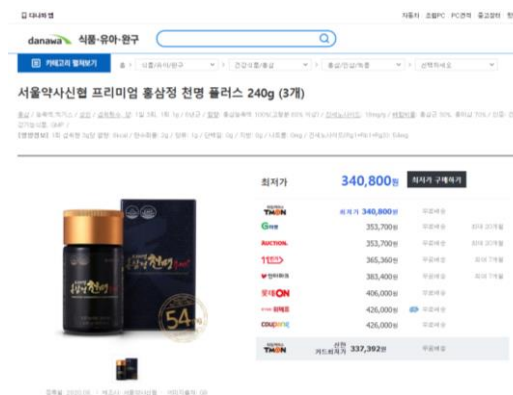
Out[239]:

	상품명	스팩목록	홍삼농축액 함량	가격	10g당 진세노 함량	10g당 가격	진세노 1mg당 가격
217	금산인삼농협 6년근 홍삼정꿀드 240g	홍삼 / 농축액,역기스 / 성인 / 섭취횟수, 양: 1일 3회 / 6년근 / 함량...	100	140000	800.0	5833	7.291250
66	서울약사신평 프리미엄 홍삼정 천명 풀러스 240g	홍삼 / 농축액,역기스 / 성인 / 섭취횟수, 양: 1일 3회, 1회 1g / 6년...	100	105750	180.0	4406	24.477778
199	삼흥 강개상인 고려홍삼정 꿀드 100g	홍삼 / 농축액,역기스 / 성인 / 섭취횟수, 양: 1일 3회, 1회 1g / 6년...	100	25200	100.0	2520	25.200000
242	명 의정 명 의정 홍삼정꿀드 120g	홍삼 / 농축액,역기스 / 성인 / 섭취횟수, 양: 1일 3회 / 6년근 / 함량...	100	60000	170.0	5000	29.411765
17	금산일품 부자 홍삼정 240g	홍삼 / 농축액,역기스 / 성인 / 섭취횟수, 양: 1일 3회 / 6년근 / 함량...	100	74760	100.0	3115	31.150000
192	풍기토종홍삼조합 장상원 홍삼정 꿀드 240g	홍삼 / 농축액,역기스 / 성인 / 섭취횟수, 양: 1일 3회 / 6년근 / 함량...	100	71030	80.0	2960	37.000000
145	중앙인삼영농조합 수홍삼사랑 6년근 홍삼정 꿀드 600g	홍삼 / 농축액,역기스 / 성인 / 섭취횟수, 양: 1일 2-3회, 1회 1g / ...	100	157520	70.0	2625	37.500000
32	금주명삼 고려홍삼농축액 250g	홍삼 / 농축액,역기스 / 성인 / 섭취횟수, 양: 1일 2회, 1회 1g / 6년...	100	38610	41.0	1544	37.658537
287	바산고려홍삼 6년근 홍삼농축액 프리미엄 240g	홍삼 / 농축액,역기스 / 성인 / 6년근 / 함량: 홍삼농축액 100%, 고형분 ...	100	64160	70.0	2673	38.185714
20	대한홍삼진흥공사 고려 홍삼농축액 VIP 100 100g	홍삼 / 농축액,역기스 / 성인 / 섭취횟수, 양: 1일 2회 / 함량: 홍삼농축액 ...	100	46400	40.0	1547	38.675000

2순위

- 진세노 1mg당 가격이 다른 제품에 비해 현저히 낮아 제품 조사 결과 다나와 사이트에서 진세노함량 기재에 문제가 있어 제외

- **약사신험 프리미엄 홍삼정 천명 플러스 240g**
- 약사신험 제품이 뽕뿌 등 커뮤니티에서 가성비 좋은 홍삼 제품으로 평판이 나 있었는데, 분석이 잘 된 것으로 보임
- 다만, 상세페이지를 클릭했을 때 가격 정보가 상이하게 나타나 분석에 한계가 있음



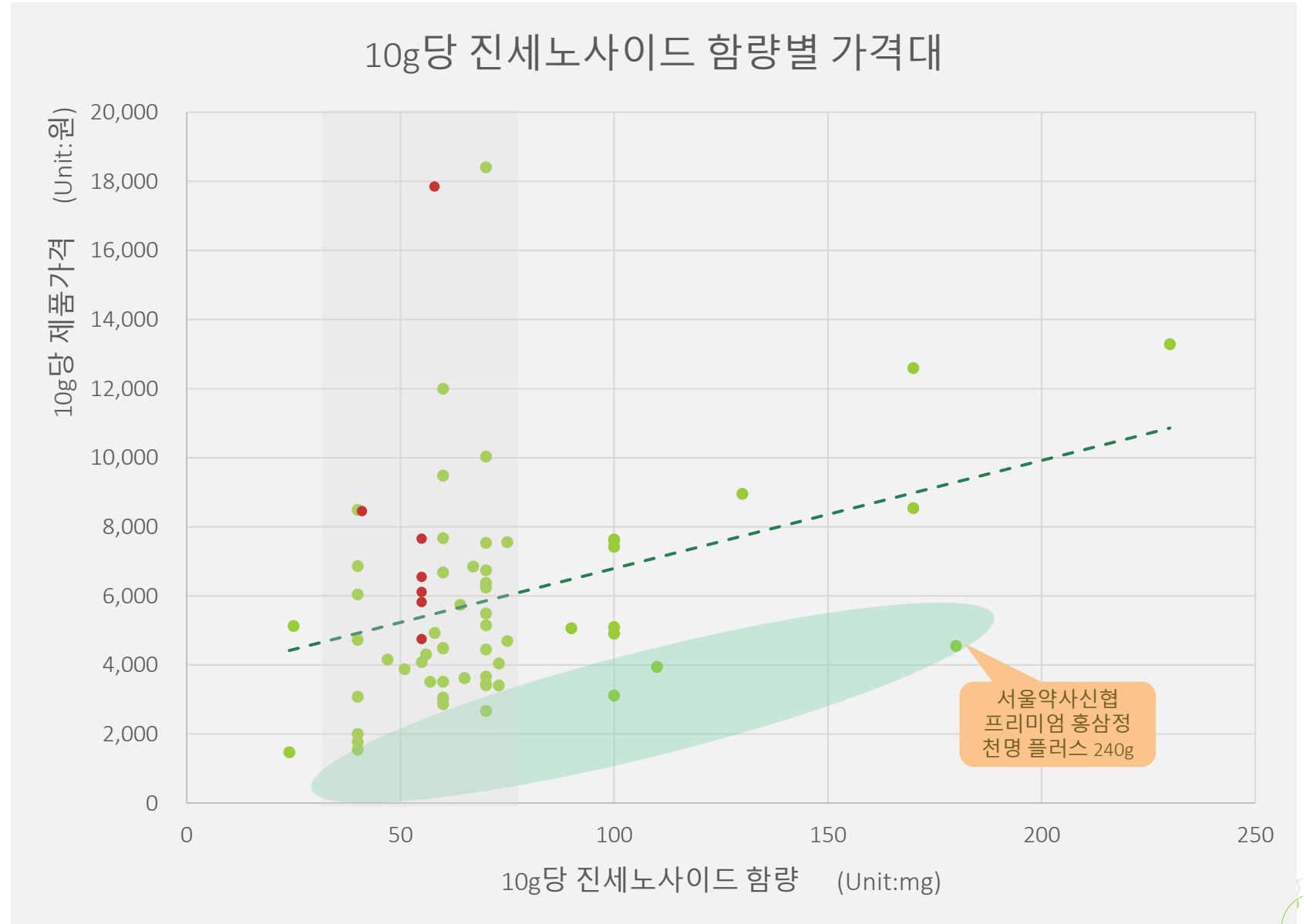
코드 소스: <https://github.com/Rachelpace/Pythonproject>

분석 결과

10g당 진세노사이드 함량 40~75mg이 대중적이며, 폭넓은 가격대를 형성함
(극단치 제외)

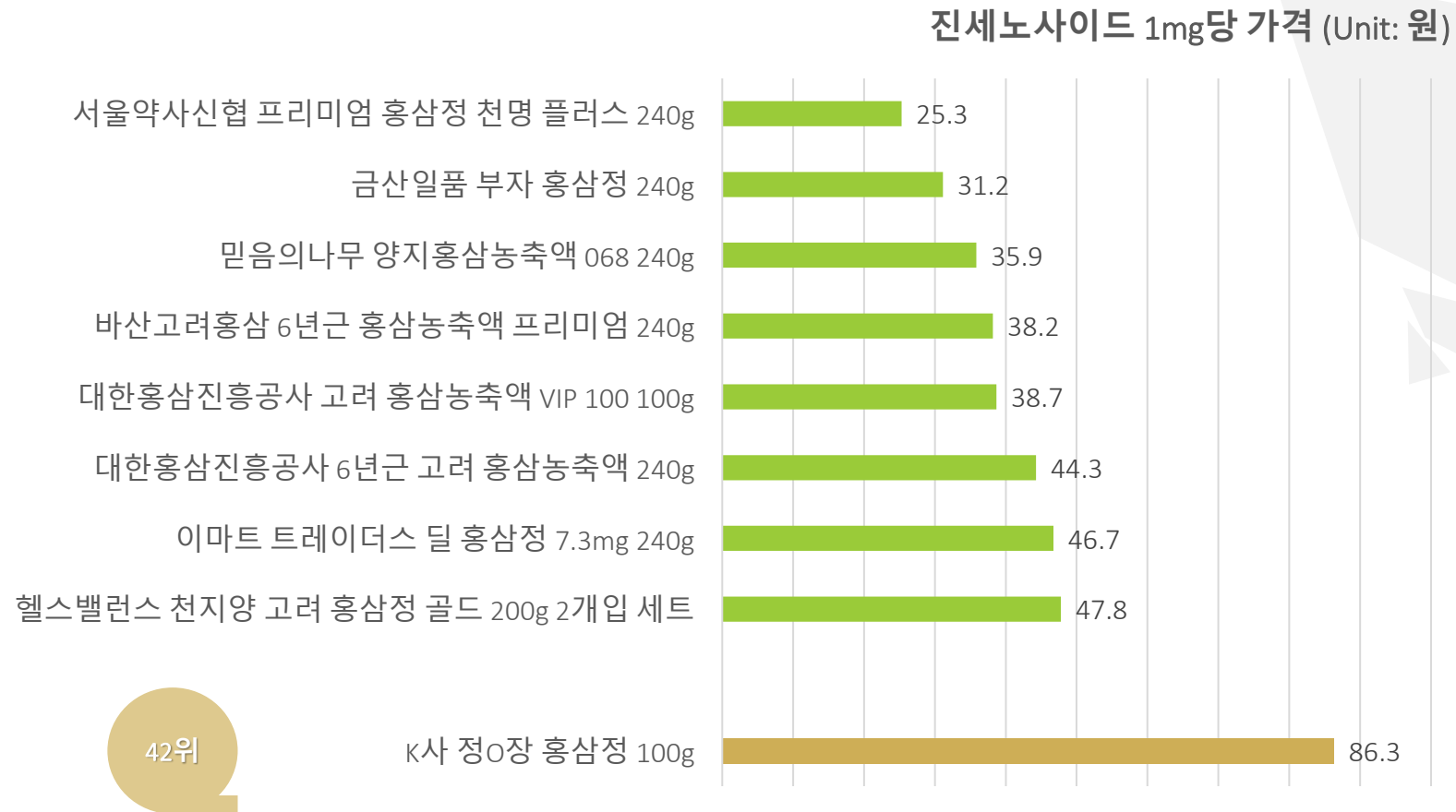
추세선에서 떨어져 평균 가격대 이하를 형성하고 있는 소수 제품들이 존재함

- 이중 서울약사신헌 제품은 높은 진세노사이드 함량에도 불구하고 저렴한 가격대로 책정된 가성비 높은 제품임
- K사 제품(Red dot)는 대부분 평균 이상의 가격대로 가성비와는 거리가 있는 편으로 분석됨



분석 결과

- 진세노사이드 1mg당 가격순으로 정렬한 결과, 중소기업제품과 이마트PB제품이 가성비가 높았음
- 이중 서울약사신험 제품이 가장 가성비가 가장 우수하며, K사 제품은 전체 67제품 중 42위 이후로 등장하고 있어 가성비가 낮은 제품으로 드러남



오류 해결 과정

크롤링 단계

```
In [5]: prod_items = soup.select('ul.product_list > li.prod_item')
len(prod_items)

Out[5]: 41
```

```
In [33]: net_items = soup.select("li[id~='productItem']")
len(net_items)

Out[33]: 40
```

- 페이지당 제품 리스트 40건을 추출해야 하는데, 리스트의 제목으로 쓰인 클래스명(prod_item product-pot)이 일부 중복됨
- Id명으로도 항목명이 걸러지지 않는 문제점 봉착

구글링을 통해 “Id가 productItem으로 시작하는” 리스트를 추출하는 코드를 작성하여 해결함
(li[id^='productItem'])

데이터 가공 단계

[illegible]

- 10g당 단가는 for 구문으로 간편히 추출했으나, 1g당 진세노사이드 함량은 for 구문을 이중으로 사용해서 추출해야 함
-For구문 추출 후 append로 리스트에 추가
- 리스트 간 나누기 연산자는 지원하지 않음

10g당 단가와 10g당 진세노사이드 함량을 새로운 컬럼으로 생성하여 새 파일로 저장한 뒤, 새 엑셀파일로 나눗셈을 한 새로운 컬럼을 생성함

데이터 분석 단계

```
In [234]: #총선농축액 합계만 따로 해서 새로운 컬럼으로 구성하고, 이중에서 100%인 행만 추출하고자 함
concentrate = []

for spec_data in data['스펙목록']:
    spec_list = spec_data.split('/')

    percentile = None

    for spec in spec_list:
        if '총선농축액' in spec:
            percent = spec.split('총선농축액')[1]
            percentile = percent.split('%')[0].strip()

    concentrate.append(percentile)
```

- 홍삼 농축액 비율값을 추출하여 새 컬럼으로 생성하였으나, 홍삼 농축액 정보가 없는 제품에도 비율이 입력되는 오류 발생

For구문 앞에 기본값을
None으로 설정한 뒤 for 구문 실행

프로젝트를 통해 깨달은 점...



깨끗한 원시 데이터

심도있는 분석을 위해서는 자체 오류가 적은 정확한 데이터를 활용하는 것이 가장 중요하다는 생각이 듭니다.



오류 처리 스킬

데이터의 다양한 오류를 예상할 수 있는 전체적인 안목이 필요함을 깨달았습니다.



의미있는 질문

데이터를 들여다 보면서 질문을 던지다 보면, 실생활에서 유용하게 활용할 수 있는 결과를 얻을 수 있을 것 같습니다.

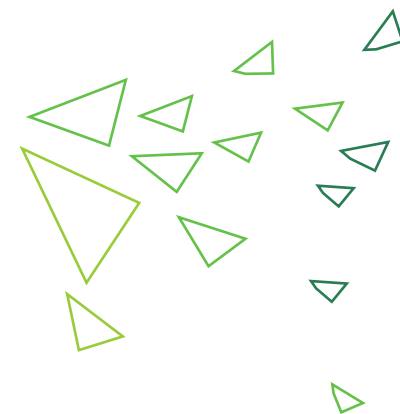


화장품 사용기한?

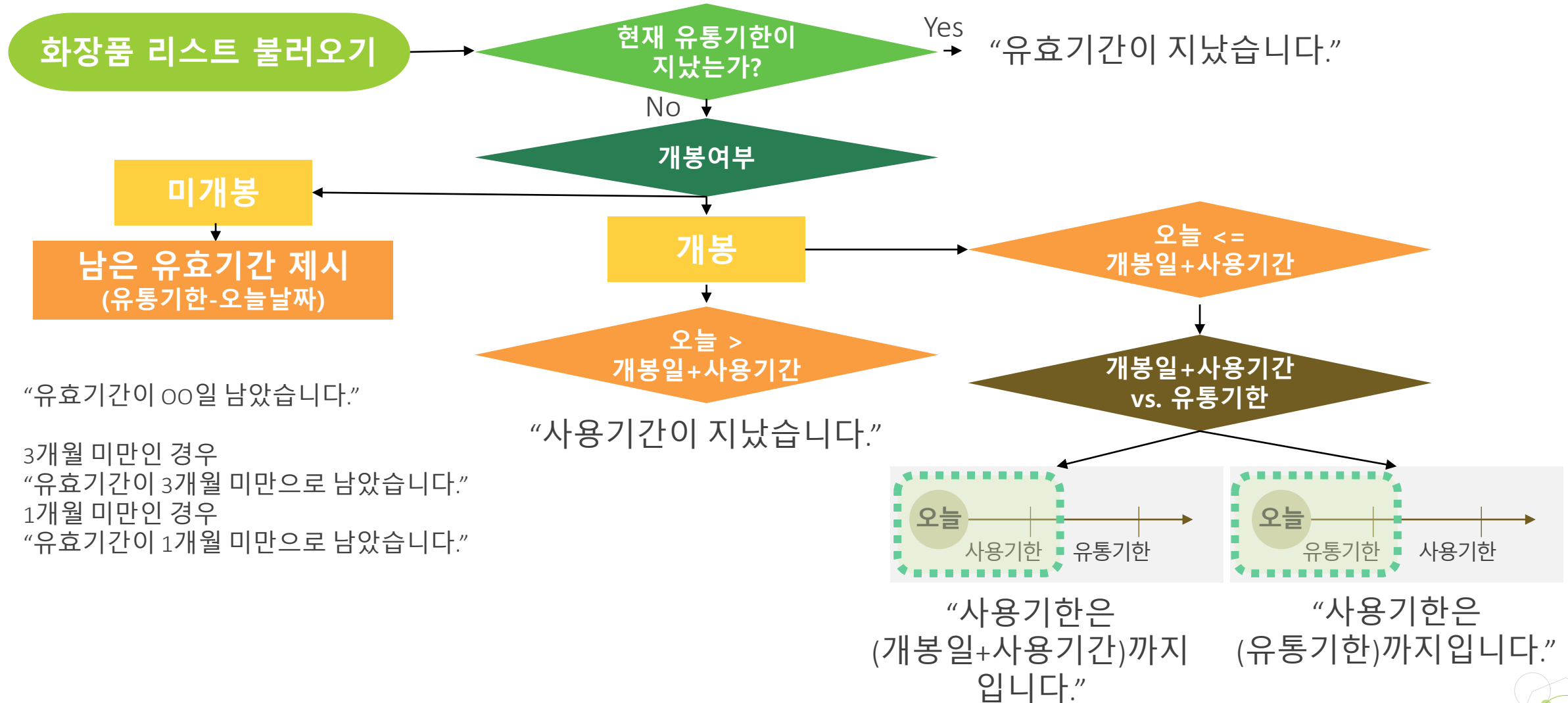
화장품의 성분에 대한 소비자들의 지대한 관심은 '안전성'에 대한 우려 때문이다. 식품에 대한 유통기한에 대해서는 민감하지만, 바르는 화장품에 대해서는 다소 둔감한 편이다. 왜 일까? 작은 화장품에 깨알 같은 글씨의 유통기한이 한눈에 들어오지 않을 뿐더러, 유통기한 정보를 읽기도 어렵기 때문이다.

#2. 화장품 프로젝트

화장대에 놓여있는 화장품만 수십가지!
유통기한도 제각각!
한눈에 유통기한을 관리할 수 있다면?

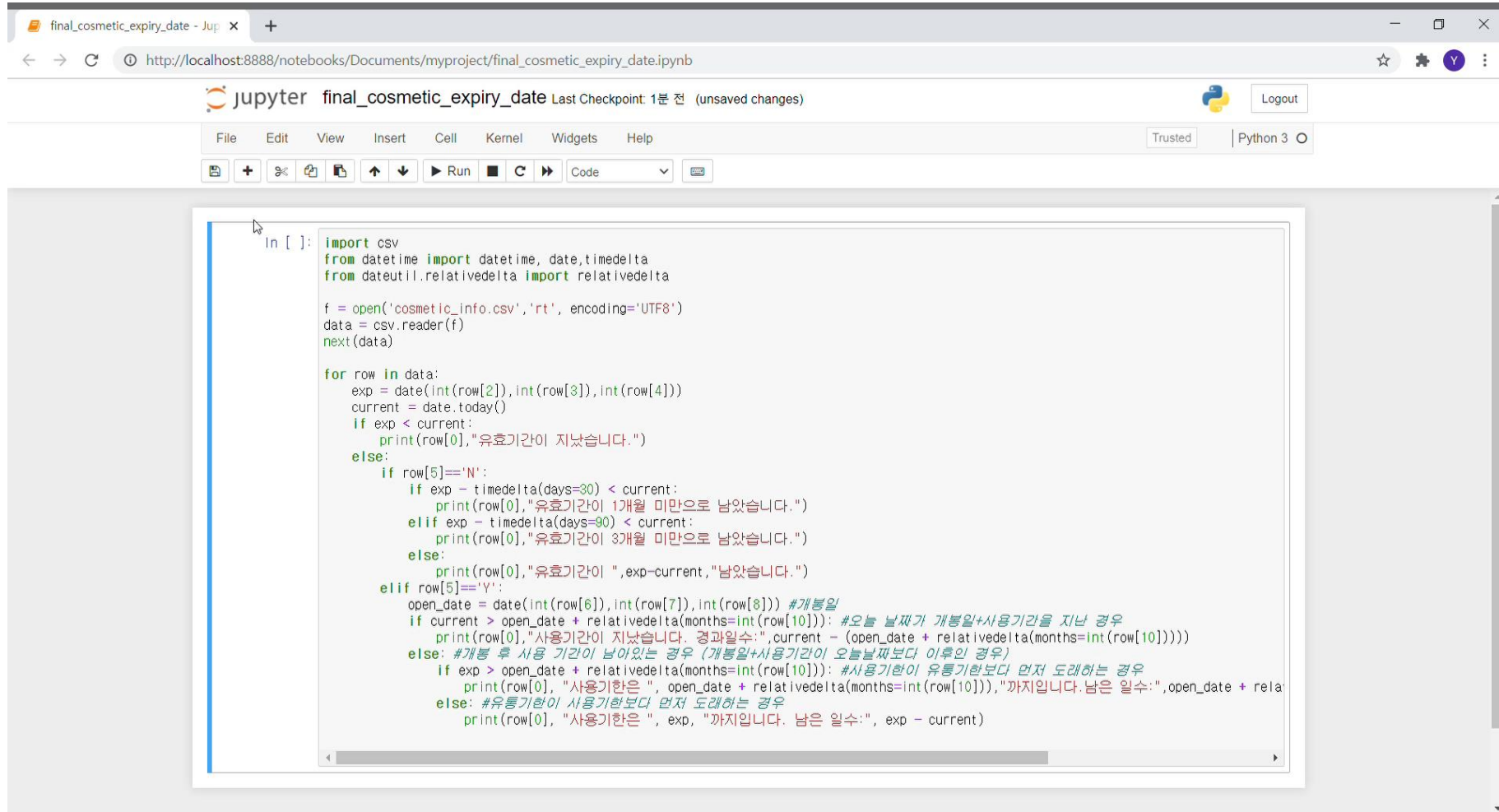


Logic Flow



- “유효기간이 00일 남았습니다.”
- 3개월 미만인 경우
“유효기간이 3개월 미만으로 남았습니다.”
- 1개월 미만인 경우
“유효기간이 1개월 미만으로 남았습니다.”

코드 실행 결과



The screenshot shows a Jupyter Notebook titled 'final_cosmetic_expiry_date' running on a local server at http://localhost:8888. The notebook contains a single code cell with the following Python code:

```
In [ ]: import csv
from datetime import datetime, date, timedelta
from dateutil.relativedelta import relativedelta

f = open('cosmetic_info.csv', 'rt', encoding='UTF8')
data = csv.reader(f)
next(data)

for row in data:
    exp = date(int(row[2]), int(row[3]), int(row[4]))
    current = date.today()
    if exp < current:
        print(row[0], "유효기간이 지났습니다.")
    else:
        if row[5] == 'N':
            if exp - timedelta(days=30) < current:
                print(row[0], "유효기간이 1개월 미만으로 남았습니다.")
            elif exp - timedelta(days=90) < current:
                print(row[0], "유효기간이 3개월 미만으로 남았습니다.")
            else:
                print(row[0], "유효기간이 ", exp - current, "남았습니다.")
        elif row[5] == 'Y':
            open_date = date(int(row[6]), int(row[7]), int(row[8])) #개봉일
            if current > open_date + relativedelta(months=int(row[10])): #오늘 날짜가 개봉일+사용기간을 지난 경우
                print(row[0], "사용기간이 지났습니다. 경과일수:", current - (open_date + relativedelta(months=int(row[10]))))
            else: #개봉 후 사용 기간이 남아있는 경우 (개봉일+사용기간이 오늘날짜보다 이후인 경우)
                if exp > open_date + relativedelta(months=int(row[10])): #사용기한이 유통기한보다 먼저 도래하는 경우
                    print(row[0], "사용기한은 ", open_date + relativedelta(months=int(row[10])), "까지입니다. 남은 일수:", open_date + rela
                else: #유통기한이 사용기한보다 먼저 도래하는 경우
                    print(row[0], "사용기한은 ", exp, "까지입니다. 남은 일수:", exp - current)
```

코드 소스: <https://github.com/Rachelpace/Pythonproject>

한계 및 개선사항

수입화장품 제조일자 정보 크롤링, 리스트 구축으로
사용자의 편의성을 극대화한 어플리케이션으로 개발



수입화장품 유통기한

수입화장품 중 배치코드(batchcode)를 통해
제조연월일을 검색해야
하는 불편함 있음

수입화장품 제조연월일
정보를 알려주는 사이
트와 연동 필요



유통기한 미표기

제조연월일만 기재된
화장품은 사용기간을
역산하여
유통기한 정보 입력

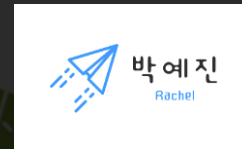



리스트 구축


브랜드, 카테고리 정보
를 리스트로 제시하면
사용자가 쉽게 데이터
입력 가능



감사합니다!



박예진 

+82 10 2560 1291 

jeanne0414@gmail.com 

<https://github.com/Rachelpace> 