



Online Cosmetics Price Comparison

GROUP 4

2018102151 ABDULLAEV AKHIDJON

2018104830 윤동주

2018102232 정주희

Project Motivation

- The percentage of cosmetics purchases online is increasing
- Cosmetics sales are expected to increase due to Post-Covid

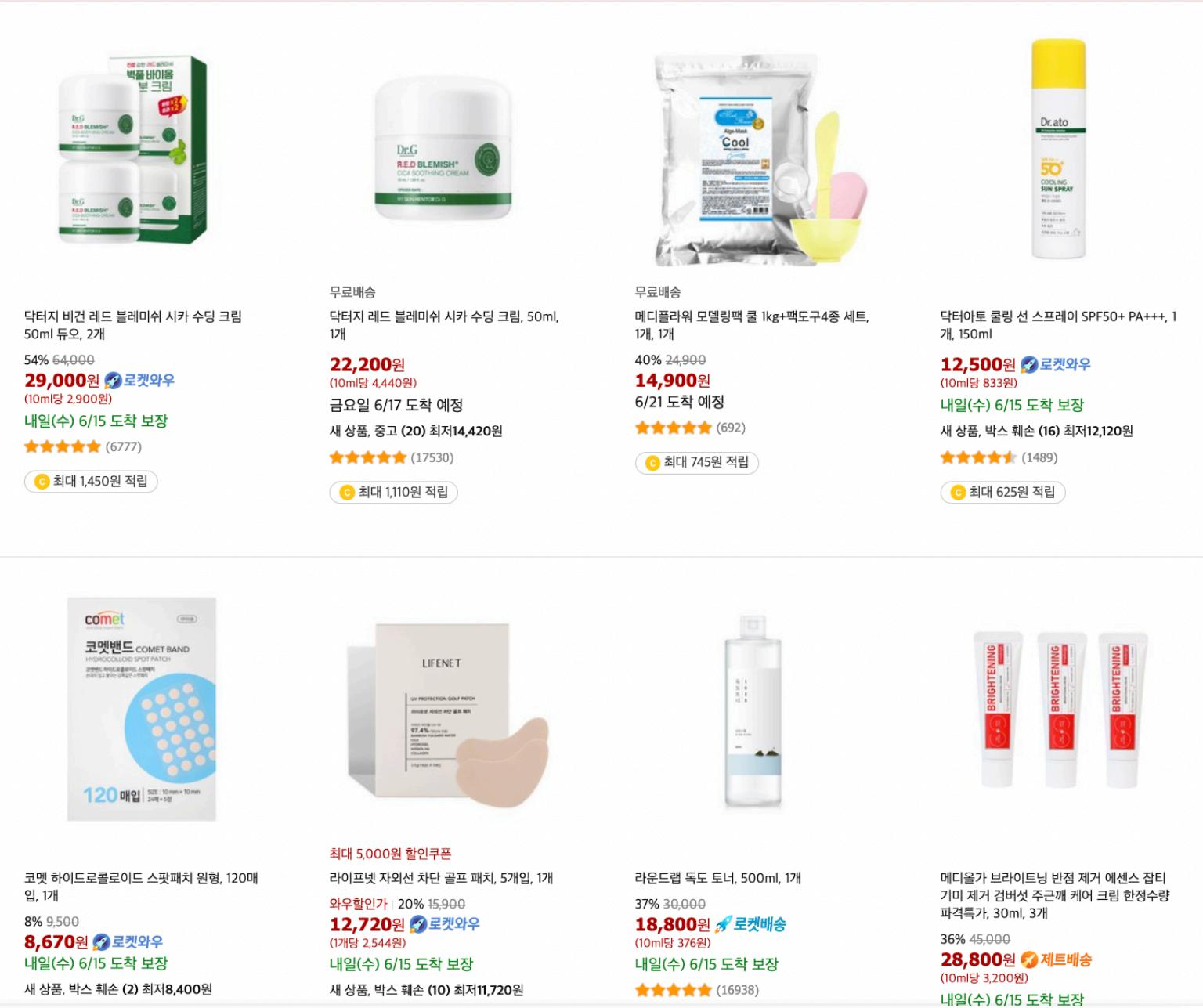


Milestone

1. Data Crawling
2. Data Storage
3. Data Analysis
4. Web Page

1. Data Crawling

Coupang

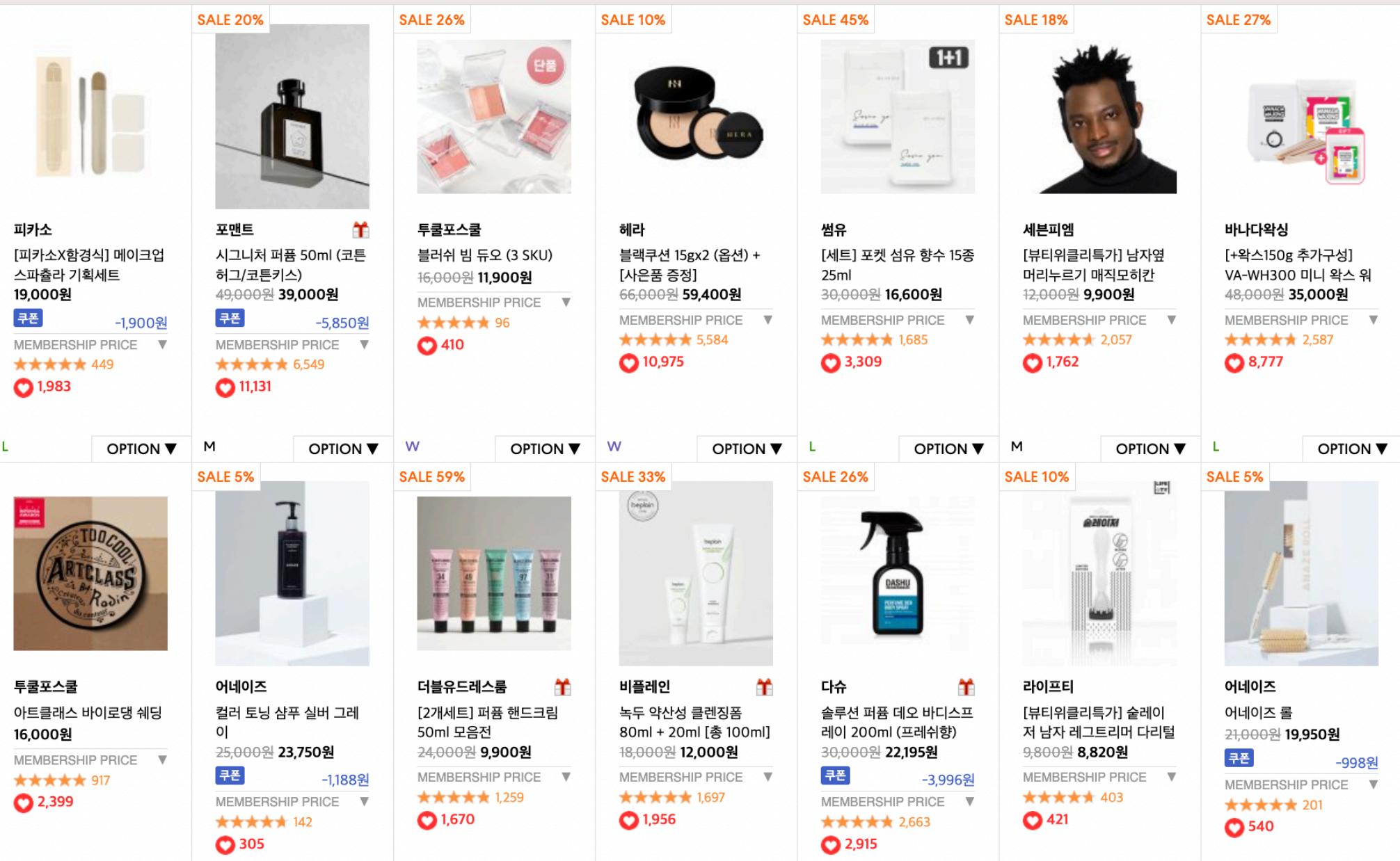


```
## 각 요소에 대해 아이디, 상품 이름, 용량, 개수 및 썸네일 이미지 파싱
for product in product_list :
    link= product.find('a', class_='baby-product-link')
    ids.append(int(link['data-item-id']))
    href.append('https://www.coupang.com'+link['href'])
    images.append('https:' + link.find('img')['src'])
    products.append(product.find('div', class_='name').text.strip())
    if product.find('del', class_='base-price'):
        origin_price = re.sub(',', '', product.find('del', class_='base-price').text.strip())
    else :
        origin_price = None
    if (product.select_one("div > span:nth-of-type(1)").text == "박스 훠손") or (product.select_one("div > span:nth-of-type(1)").text == "중고"):
        dis_price = re.sub(',', '', product.select_one("div > strong:nth-of-type(2)").text.strip())
    else:
        dis_price = re.sub(',', '', product.find('strong', class_='price-value').text.strip())
    old_prices.append(origin_price)
    new_prices.append(dis_price)
```

- Get all product from product list
- Parse data from product

1. Data Crawling

Musinsa



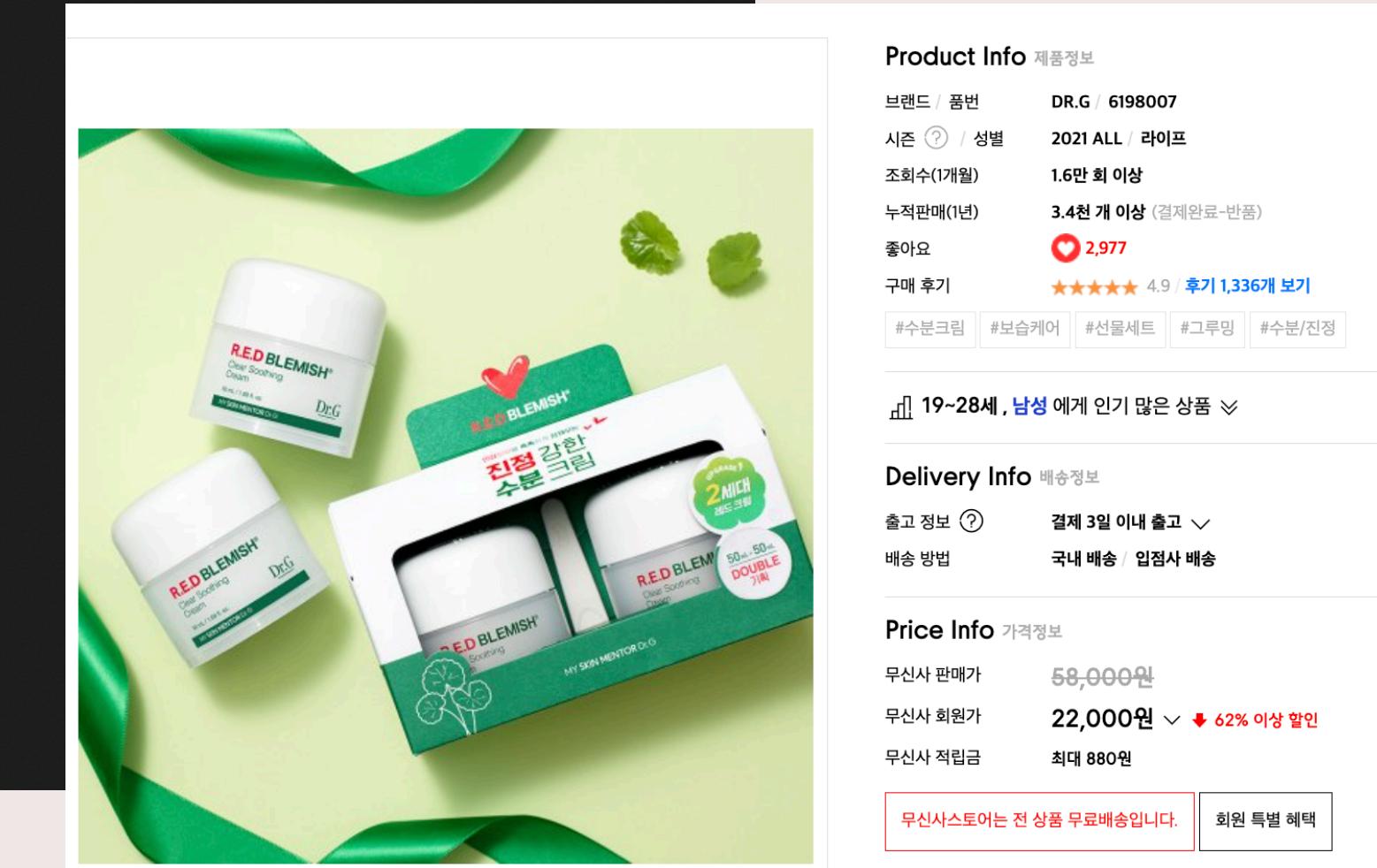
```
"links": [  
    "https://www.musinsa.com/app/goods/1755093",  
    "https://www.musinsa.com/app/goods/1850705",  
    "https://www.musinsa.com/app/goods/1738489",  
    "https://www.musinsa.com/app/goods/1584811",  
    "https://www.musinsa.com/app/goods/2196195",  
    "https://www.musinsa.com/app/goods/2517044",  
    "https://www.musinsa.com/app/goods/2252220",  
    "https://www.musinsa.com/app/goods/2477086",  
    "https://www.musinsa.com/app/goods/2561926",  
    "https://www.musinsa.com/app/goods/1017014",  
    "https://www.musinsa.com/app/goods/2196101",  
    "https://www.musinsa.com/app/goods/1226498",  
    "https://www.musinsa.com/app/goods/2196075",  
    "https://www.musinsa.com/app/goods/1911807",  
    "https://www.musinsa.com/app/goods/2196150",  
    "https://www.musinsa.com/app/goods/1599736",  
    "https://www.musinsa.com/app/goods/1606517",  
    "https://www.musinsa.com/app/goods/1785353",  
    "https://www.musinsa.com/app/goods/1471574",  
]
```

- Get all link from product list

1. Data Crawling

Musinsa

```
def scrapePage(link):
    l = link
    try:
        sleep(0.5)
        url = requests.get(link, headers = {'User-Agent': 'Mozilla/5.0'})
        print(url.status_code)
        soup = BeautifulSoup(url.content, "html.parser")
        title = soup.find("span", {"class": "product_title"}).get_text().strip()
        category = soup.find("div", {"class": "product_info"}).find_all("a")[1].get_text()
        brand = soup.find("ul", {"class": "product_article"}).find("li").find("p", {"class": "product_article_contents"}).get_text().split()[0]
        image = soup.find("div", {"class": "product-img"}).find("img").get("src")
        sale = '0'
        try:
            sale = soup.find("span", {"class": "txt_kor_discount"}).get_text().strip()[:2]
            old_price = soup.find("span", {"class": "product_article_price"}).get_text().strip()
            new_price = soup.find("span", {"class": "txt_price_member"}).get_text().strip()
        except Exception as e:
            old_price = soup.find("span", {"class": "product_article_price"}).get_text().strip()
            new_price = soup.find("span", {"class": "product_article_price"}).get_text().strip()
        try:
            optionData = soup.find("select").find_all("option")
            del optionData[0]
            option = []
            for k in optionData:
                option_data = {
                    "option": k.get('value'),
                    "stock": k.get('jaego_yn')
                }
                option.append(option_data)
        except Exception as e:
            option = []
    
```



- Get all link page from requests
- Parse data from detail pages

1. Data Crawling

Musinsa - Optimization

- Original Code took 7-8 hours to get all data
- Improvement of Time by MultiThreading

```
if __name__ == "__main__":
    starttime = time()
    filename = "data.json"

    html = requests.get("https://www.musinsa.com/category/015", headers = {'User-Agent': 'Mozilla/5.0'})
    bsObject = BeautifulSoup(html.text, "html.parser")
    categories = [x.find('a').get('data-code') for x in bsObject.find("div", {"id": "category_2depth_list"}).find_all("li")]

    links = []
    #Collecting products link to links list:
    """with closing(Pool(cpu_count())) as pool:
        items = pool imap_unordered(getallProductLinks, categories)
        for i in items:
            links.extend(i)"""

    with open("link.json", 'r') as file:
        links = json.load(file)

    with closing(Pool(cpu_count())) as p:
        products = p.map(scrapePage, links['links'])
        write_json(products)

    print("time:", time()-starttime)
```

1. Data Crawling

Olive Young

카테고리	오톡	신상	랭킹	프리미엄관
뷰티				
스킨케어 >	더모 코스메틱 >	헤어케어 >		
스킨/로션/물인원	스킨/로션	샴푸/린스		
에센스/크림	에센스/크림	트리트먼트/팩		
미스트/오일	미스트/오일	헤어기기		
마스크팩 >	마스크팩	헤어브러쉬		
시트팩/패드	클렌징	염색약/펌		
페이셜팩	선케어	스타일링		
코팩/매치	바디케어	탈모케어		
클렌징 >	메이크업 >	에센스		
클렌징폼/젤	립메이크업	향수/디퓨저 >		
오일/워터/리무버	베이스메이크업	여성향수		
필링/패드	아이메이크업	남성향수		
선케어 >	네일 >	캔들/디퓨저		
선블록	플리쉬	미용소품 >		
태닝/애프터션	팁/스티커	페이스		
	반경화	아이		
	케어	헤어/바디		
바디케어 >	코튼	코튼		
워시/스크럽	디바이스	디바이스		
로션/오일	네일	네일		
제모용품	기타소품	기타소품		
핸드케어	남성 >			
립케어	스킨케어			
바디미스트	바디케어			
데오도란트	헤어케어			
풋케어	메이크업			
선물세트	쉐이빙			
	향수			

- Issue

- Categories can only be obtained by clicking.

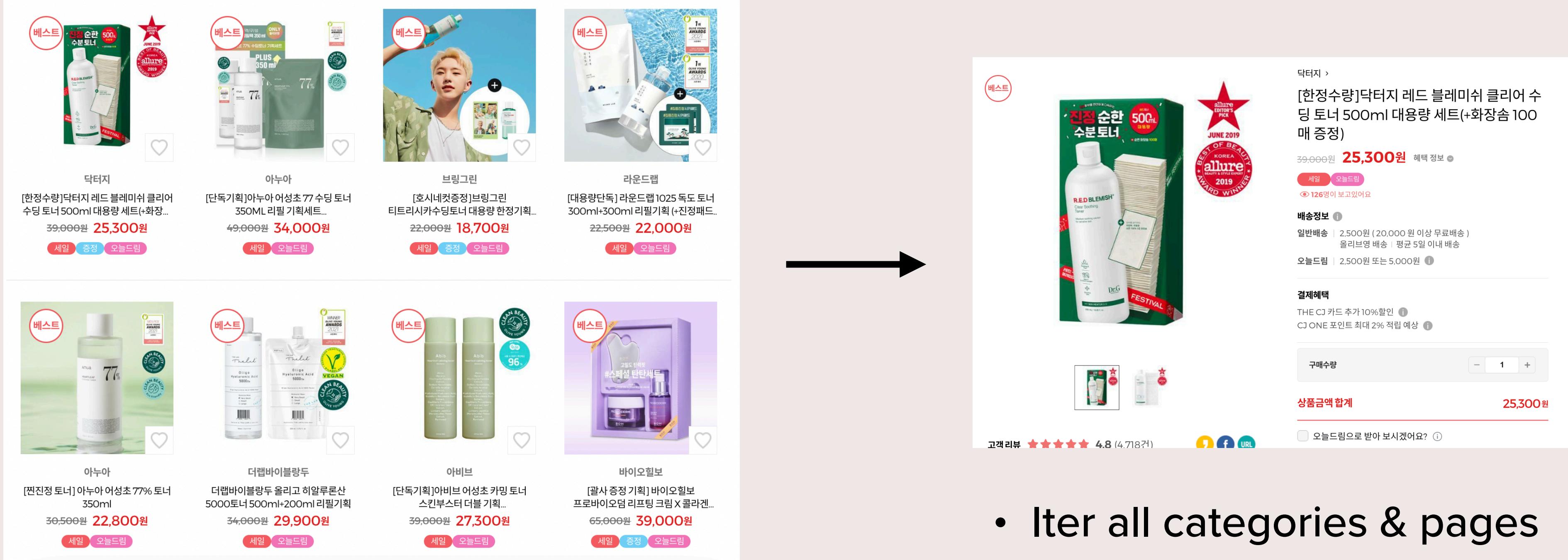
- Selenium takes too much time

- Solution

- Obtain category data by Selenium first, get rest of the product data by requests

1. Data Crawling

Olive Young



- Iter all categories & pages
- Get product info from detail pages

1. Data Crawling

Daily Crawling

- Using ec2 Instance as a crawling server
- Daily crawling using crontab

```
0 0 * * * python3 /home/musinsa/app.py
```

```
0 0 * * * python3 /home/ubuntu/olive_crawl/olive_crawl.py
```

```
0 0 * * * python3 /home/coupaang/crawl.py
```

1. Data Crawling

Daily Crawling

- Result

- Coupang 43MB
- Musinsa 110MB
- Olive Young 112MB
- Total 265MB

```
musinsa@ip-172-31-32-254:~/cr$ ll
total 111992
drwxrwxr-x 2 musinsa musinsa    4096 Jun 14 11:41 .
drwxr-xr-x 5 musinsa musinsa    4096 Jun 14 11:41 ../
-rw-rw-r-- 1 musinsa musinsa 8860880 Jun  2 03:38 20220602.json
-rw-rw-r-- 1 musinsa musinsa 8856584 Jun  3 03:41 20220603.json
-rw-rw-r-- 1 musinsa musinsa 8856533 Jun  4 03:36 20220604.json
-rw-rw-r-- 1 musinsa musinsa 8856437 Jun  5 03:36 20220605.json
-rw-rw-r-- 1 musinsa musinsa 8856287 Jun  6 03:38 20220606.json
-rw-rw-r-- 1 musinsa musinsa 8852324 Jun  7 03:39 20220607.json
-rw-rw-r-- 1 musinsa musinsa 8815514 Jun  8 03:38 20220608.json
-rw-rw-r-- 1 musinsa musinsa 8811488 Jun  9 03:37 20220609.json
-rw-rw-r-- 1 musinsa musinsa 8788028 Jun 10 03:41 20220610.json
-rw-rw-r-- 1 musinsa musinsa 8799436 Jun 11 03:39 20220611.json
-rw-rw-r-- 1 musinsa musinsa 8729128 Jun 12 14:46 20220612.json
-rw-rw-r-- 1 musinsa musinsa 8739459 Jun 13 08:32 20220613.json
-rw-rw-r-- 1 musinsa musinsa 8812691 Jun 14 03:45 20220614.json
musinsa@ip-172-31-32-254:~/cr$
```

1. Data Crawling

Olive Young - Improvement

- Lack of Data
 - Information data Combined - 265MB
 - Additional 500MB
- Lack of Analysis
 - Spark not only for preprocessing

1. Data Crawling

Olive Young - Improvement

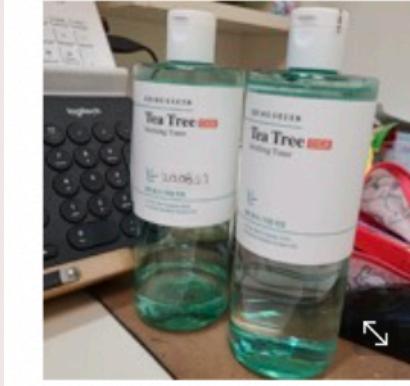
- Additional Crawling
 - Review Data
 - Maximum 30 Pages of Review
 - Using Selenium
 - Using Multiple Crawler in different categories
 - Total 582M of review data

★★★★★ 2021.02.13 · 매장

피부타입 복합성에 좋아요 | 피부고민 진정에 좋아요 | 자극도 자극없이 순해요

착한 진정-수딩 토너는 브링그린 티트리 시카 토너라고 쓴다.
*재구매지수: 이미 재구매 함 / 추천 피부타입: 모든 피부타입(건성은 이후 기초단계는 당연히 탄탄하게 할 거라고 생각하였습
니다. 토너다 보니 단독 사용하는 경우는 없겠죠?!)

저는 마스크뿐만 아니라 평소 메이크업때문에 딥클렌징을 해서인지 세안 후 붉게 열감이 오르는 타입이라 진정에 도움
되는 토너를 찾던 중, 향기롭고 달콤한 티트리 토너를 찾았습니다! 다행히도 좋지만 아침 or
저녁에 간단하게 스킨팩 할 때 촉촉하게 헹구는 마무리감이 좋습니다! 더불어 가장 좋았던 건, 냉장고에 굳이 넣
지 않아도 열감을 바로 DOWN 시킬 수 있게 시원한 쿨링감이 느껴져서 좋아요! 그러다 보니 트러블 때문에 항상 열감을 얼굴
에 달고 다니시는 분들에게 더 도움되지 않을까 싶긴하네요! 더불어 대용량이라 일단 옆에 두면 든든! 대용량이라 이번에
는 토너패드처럼 공병패드통에 화장솜 넣어 토너패드처럼 사용해볼까요! 지금까지 스킨팩 or 닦는 무리없이 만족하
며 사용했으니 패드로도 좋을 것 같아요! 다만 aha 성분이 들어 있어 저자극으로 각질케어가 가능할 거 같았는데 그거에 대
한 효과는 크게 체감하지는 못했지만 잘 되고 있겠죠..?



2. Data Storage

HDFS

- All product information file is in hdfs://input

```
hadoop@master:~ $ ./hadoop-2.7.6/bin/hdfs dfsadmin -report
Configured Capacity: 95063789568 (88.54 GB)
Present Capacity: 81792978135 (76.18 GB)
DFS Remaining: 80798461952 (75.25 GB)
DFS Used: 994516183 (948.44 MB)
DFS Used%: 1.22%
Under replicated blocks: 276
Blocks with corrupt replicas: 0
Missing blocks: 0
Missing blocks (with replication factor 1): 0
```

3. Data Analysis

Using Pyspark

- All
 - Combine all crawling files
 - Parse product name and get volume using regular expression
 - In pyspark, regexp_extract & regexp_replace
 - Calculate discount percentage & cost effective
 - Remove unnecessary column
 - Unify the column names

3. Data Analysis

Using Pyspark

- Issue
 - sc.read.json -> Data is null
 - json.load file and sc.parallelize
(Same problem occurred in musinsa data)
 - Write Json
 - If some data is null, exception occurs
 - Solved by adding default data
 - With function lit()

3. Data Analysis

Using Pyspark

- Review

A000000166675	#이것이 진정이다#여드름너나와!#자녀피부구원템 이 제품은 1년전부터 스멀스멀 뾰루지가 올라오고 있던 자녀를 위해 구매하기 시작했습니다. 사춘기 뾰루지엔 어성초가 좋다는말이 세간에 퍼져있을 때 우선 저는 구* 어성
A000000166675	복합성 피부에 많이 민감한 피부입니다. 요즘 봄 바람에 아침 저녁으로 기온차가 심해서 그런지 홍조도 올라오고 특히 볼쪽이 많이 건조해서 화장을해도 들뜨고 많이 건조함을 느낍니다. 마스크 때문인지 턱과 볼부분에 트러
A000000166675	날씨 더워지면 광대가 뜨거워지면서 내 얼굴이 호빵면이 되어가는 것 같고 좀 더 더워지면 졸살이 심해지는 30대 악건성,민감,모공부자 졸살이 쉽게 올라오는 여자예요! 계절 바뀌면 제가 딱 이래요.이미 핫템인 아누아 어성초
A000000166675	이름부터 어성초77% 수딩 토너인 만큼 피부가 뺄갛게 달아오르거나 트러블이 올라왔을때 사용을 해보았어요. 피부진정에 효과적이더라고요. 특히 수딩 토너라서 피부진정 후 보습감도 느껴져서 촉촉했어요. 복합성 피부에:
A000000166675	아누아 제품은 꾸준히 써오고 있는 편편팬입니다~~~ 올리브영 랭킹보면 아누아 토너는 항상 랭킹 최상단에 있더라구요 이번에 세트가 새로 나왔길래 바로 구입했습니다.. 역시나 후기도 너무 좋고 요즘엔 주위에 물어보니 지인
A000000166675	원래 민감성피부에 약간의 건성 피부를 가지고 있었는데 코로나 시작이후 마스크를 쓰면서 어느날 갑자기 접촉성 피부염인지 짜지도 않는 붉고 아프게 오돌토돌하게 올라와서 엄청나게 스트레스를 받고 있었습니다. 유튜브
A000000166675	😊 순한 첫느낌아누아 브랜드는 첫사용입니다인기가 좋아 최근 스킨 브랜드에서 당당히 1위에 있길래 그냥 넘어갈수없어 바로 구매했어요😊 약모밀축출물 77프로들어있어요 어성초하면 진정으로 유명하잖아요 순~~~한 느
A000000166675	📌 아누아 어성초 토너 리뷰📌 자주 사용하는 토너 중 하나인데 또 구매했네요ㅎㅎ아침 저녁으로 세안 후 사용하는데 확실히 진정시켜주는 효과가 있고 말끔히 잘 스며드네요~~~😊 특히 저녁에 피로한 피부와 스트레스상 :
A000000166675	약 2주간의 사용을 하고 리뷰를 남기려 왔어요~~ ✓우선 저의 피부 타입은 수부지로 어드름 흉터와 예민성을 가진 피부인데요! 요즘 화장하고 마스크끼고 하니까 홍조까지 덤으로 생겼더라구요 ㅠㅠ 그래서 얼굴에 열감증 낮
A000000166675	요즘 피부가 너무 건조하고 턱부분에 졸쌀 여드름이 올라와서 어성초 비누를 쓰는 중인데 어성초가 가장 많이 들어있는 토너길래 출린듯이 선택했어요 역시 피부 진정과 트러블에 좋은 어성초가 절반이상이나 들어있어서 :
A000000166675	요즘 어성초 상품이 유행하고 많이들 쓴다고 하는데 이렇게 체험단 통해서 받아서 써보니까 웨들 그러는지 알겠더라구요! 이마에 예전에 졸쌀여드름이 조금 난었는데 마스크싸면서 이제는 코옆이나 이런데 블랙헤드랑 졸쌀이
A000000166675	기초제품 유독민 아누아 어성초 스킨으로 정착했습니다원래 쓴던 스키인데 울영체험단에 끄길래 바로 신청했어요진짜 사용해본 스키중에 가장 잘쓰고 있습니다특히 스킨팩 할 때 좋습니다어드름성 제 피부에 효과 있길래 랭
A000000166675	제 인생템 아누아 브랜드 가장 유명한 스키에 대해서 후기를 써보려고합니다 많은 도움이 되길 바라요!!좋은점 ✨스킨팩하기 너무 좋다 저는 스킨팩을 정말 많이 하는데요 스킨팩하기에 딱 최적화된 스키이다 라고 생각합니다
A000000166675	💧 유분감 없이 수분으로 가득찬 어성초 토너! 토너, 흡토 어떤 형태에도 다 편해요트러블 진정은 솔직하게 쏘쏘였지만(붉게 올라온 부분이 있어서 흡토+토너팩 했지만 바로 마법처럼 진정되고 가라앉는건 솔직하게
A000000166675	😊총평: 토너보다 앰플 추천! 토너는 진정, 보습에 좋았고 앰플도 진정 보습에 더더 좋았지만 지성분들께는 앰플이 리치할 수 있어요 그 점만 주의하면 굿!(건성/민감성/졸쌀여드름) 모두가 목놓아 소리치는 아누아!! 그렇:
A000000166675	제 피부타입은 민감하고 예민하며 홍조 여드름이 있으며 생활패턴에 피부가 영향을 많이 받아요저는 웬만하면 귀찮아 토너는 지금까지 많이 사용해보지못하고 매일 패드로 닦아내다보니 피부가 너무 민감해져 요즘에는 미스
A000000166675	* 민감성/복합성- 제형(★★★) 균적이지 않고 물처럼 흐르는 제형입니다. 흐르는 제형이라 빠르게 흡수돼요! 토너팩이나 닦토율으로 사용하면 좋은 제형입니다.- 수분력/보습력(★★★) 물 같은 제형인데도 피부에 수분을 꽉
A000000166675	어성초는 피부 염증과 해독에 효능이 탁월하다고 알려져 있죠. 여드름이나 아토피 피부에 비효과적 효과가 있다고들 하더라구요. 민감성이고 건성인 제 피부에도 잘 맞는지 사용해보고 있습니다. 일단 가장 맘에 들었던 건 용량!
A000000166675	울영인점 전에도 네이버쇼핑에서 광고 해서 알고 있었던 토너예요! 우연히 공홈 들어갔다가 5000개? 넘는 리뷰 보고 광고인지 아닌 진짜 넘 좋아서 그런지 헷갈렸지만 좀 흑해서 나중에 토너 바꿀때 써봐야지 했었는데 울영 :
A000000166675	아주 순한 진정효과에 특화된 제품이에요어성초가 여드름 진정에 참 좋다하는데이 제품은 어성초 함유량이 77프로 들어있고 물보다 많이들어있어요알코올성분이 없어서 피부에 자극이 전혀 없는굉장히 순한 토너예요전에 0
A000000166675	졸쌀 여드름에 좋다고 하도 순위권에 올라와있고, 광고를 많이 해서 구매해봤어요.저는 여드름이 볼쪽에 조금있고, 이마에는 졸쌀 여드름이 많아서 상당히 고민이었거든요, 마스크를 쓰다보니 이마가 위주로 보이는데, 항상 신
A000000166675	스트레스 받으면 트러블성이고평소에는 건성인 피부인데어성초가 민감성 피부진정에 좋다?라는 말을 듣고 유튜브 보고 구매!-제형제형은 흐르는 물 같은 제형이에요토너가 진짜 맑고 거버워요! 0 0 0 -보습감/수분감을 같아서
A000000166675	ଓ랫동안 유세* 토너를 사용—대략 10년 정도? 😊➡️ 기본적으로 민감 + 건성 피부 / 이전에 아토피 알음환절기엔 국소부위만 다소 유분기 있는 피부유형 ➡️ 주로 유세* 토너 사용하다중간에 라로슈포* 토너를 교차 사용
A000000166675	두통째 쓰고 있는 토너예요 아이들 멤버가 사용해서 유명해졌다고 하는데, 저는 그 이후에 올리브영 랭킹에서 발견하고 어성초가 잘 맞기도 하고 리뷰도 좋아서 구매해봤어요 너무 잘 맞아서 데일리 토너로 아침, 저녁 사용하고
A000000166675	어성초가 트러블 진정에 좋다해서 여름에 사용하려고 구입했어요~^^아누아가 어성초로 유명하던데.. 아누아가 한 3달 전 울영에 입점되고 특가로 19.900원에 판매할 때 갯했어요ㅋㅋ 아직 사용하고 있는 토너가 있어 좀 더

• OBJECTIVE
: Extract the most frequently used words in the reviews for each product

3. Data Analysis

Using Pyspark

- Review

```
comment = comment.withColumn("revised_comment", regexp_replace("comment", "[^A-Za-z0-9가-힣]", ' ')).drop('comment')

sq = SQLContext(sc)

comment.createOrReplaceTempView("comment")
merged = sq.sql("SELECT id, concat_ws(' ', collect_list(revised_comment)) as comment FROM comment group by id")

tokenized = merged.withColumn("comment", split("comment", " "))

tokenized.printSchema()

root
 |-- id: string (nullable = true)
 |-- comment: array (nullable = false)
 |    |-- element: string (containsNull = true)

token_count = tokenized.select("id", explode("comment").alias("token"))\
    .groupBy("id", "token")\
    .count()\
    .groupBy("id")\
    .agg(collect_list(struct(col("token"), col("count"))).alias("text"))
```

- Remove Special characters
- Split review
- Count words

3. Data Analysis

Using Pyspark

- Review

```
def def_sort(x) :
    return sorted(x, key=lambda x :x[1], reverse = True)

udf_sort = udf(def_sort, ArrayType(StructType([StructField("token", StringType()),StructField("count",LongType())])))

sorted_t = token_count.select("id", udf_sort(col("text")).alias("text"))
```

```
option = ["저는", "제품", "선물을", "이건", "저의", "저희", "요즘", "본인", "원래", "했습니다", "이다", "사용", "후기", "이거", "좀", "잘", '']
stop_words = olive.rdd.flatMap(lambda x : [x.brand, x.big_category, x.small_category]).collect()
stop_words.extend(option)

is_none = lambda x : x != ''
is_stop = lambda x : ~(x.isin(stop_words))

res = sorted_t\
    .withColumn("removed", filter(col("text.token"), is_none))\
    .withColumn("token", filter(col("removed"), is_stop))\
    .withColumn("token", slice(col("token"), 1, 10))\
    .drop("text")\
    .drop("removed")
```

```
joined = res.join(olive, res.id == olive.id)
joined = joined.dropDuplicates(['id'])
final = joined.select(['name','comment.id', 'token'])
```

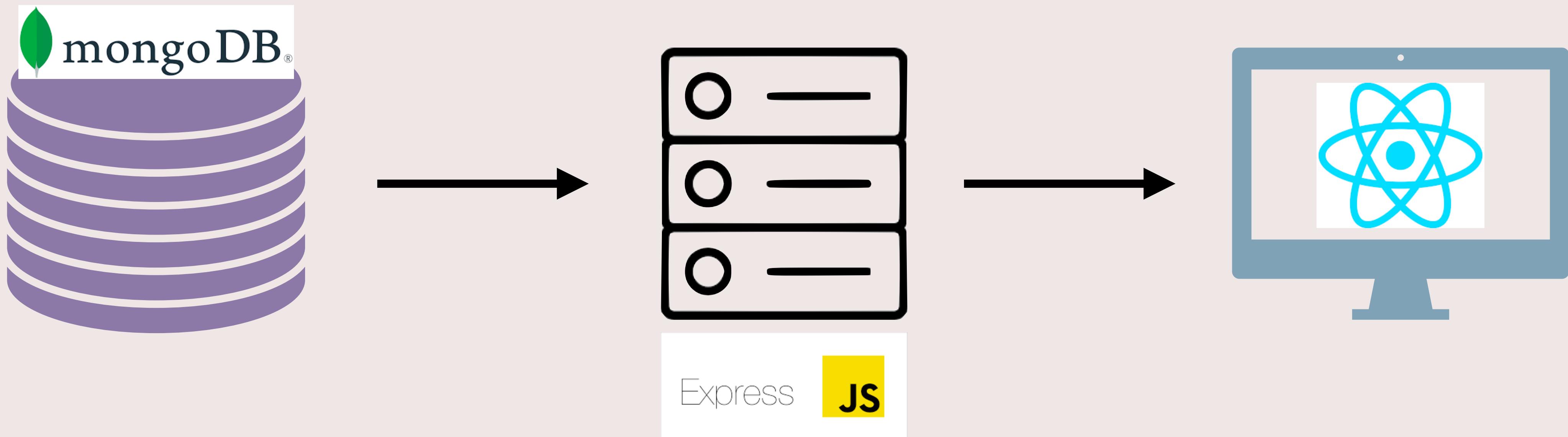
- Sort by count
- Remove blank word
- Remove brand or category name
- Remove stop words

3. Data Analysis

Using Pyspark

- Spark-Submit
 - Master yarn
 - Issue : yarn memory
 - Deploy mode
 - Cluster
 - Executor-memory : 512m
 - Driver-memory : 512m

4. Web Page



4. Web Page

- MongoDB

```
from pymongo import MongoClient

client = MongoClient("mongodb://[REDACTED]",authSource="admin")
db = client.bigdata
productDB = db['products']
categoryDB = db['category']
commentDB = db['comments']
priceDB = db['price']

with open("olive_category.json") as f :
    category = json.load(f)

categoryDB.insert_one(category)

<pymongo.results.InsertOneResult at 0x14d12f430>

js = df.toJSON().map(lambda x : json.loads(x)).collect()

productDB.insert_many(js)
```

```
price_js = olive_price.toJSON().map(lambda x : json.loads(x)).collect()

priceDB.insert_many(price_js)
```

- Export data frame after converting to JSON

☰ K-Beauty



Sea view, infinity pool, spa bath

My Account

스킨케어

마스크팩

클렌징

선케어

더모 코스메틱

메이크업

네일

바디케어

헤어케어

향수/디퓨저

미용소품

남성

34000₩ 49000₩
아누아 어성초 77 수딩 토너
350ML 리필 기획세트

19800₩ 22500₩
라운드랩 1025 독도 토너
300ml+300ml 리필기획

29900₩ 34000₩
더랩바이탈루올리고 히알루론
산 500ml+200ml 리필기획

16900₩ 22000₩
브링그린 티트리시카수딩토너 대
용량 한정기획

23400₩ 39000₩
닥터지 레드 블레미쉬 클리어 수
딩 토너 500ml 대용량 세트

14630₩ 22000₩
넘버즈인 3번 결광가득 에센스 토
너 200ml

18900₩ 28000₩
아이디얼포맨 퍼펙트올인원밀크
한정기획_100mL증정

27300₩ 39000₩
아비브 어성초 카밍 토너 스킨부
스터 더블 기획

24800₩ 31000₩
에스트라 아토베리어365 로션
150ml 장벽작공 기획

17900₩ 29000₩
닥터지 레드 블레미쉬 포 맨 올인
원 플루이드

17900₩ 29000₩
닥터지 레드 블레미쉬 포 맨 올인
원 플루이드

17900₩ 29000₩
닥터지 레드 블레미쉬 포 맨 올인
원 플루이드

☰ K-Beauty

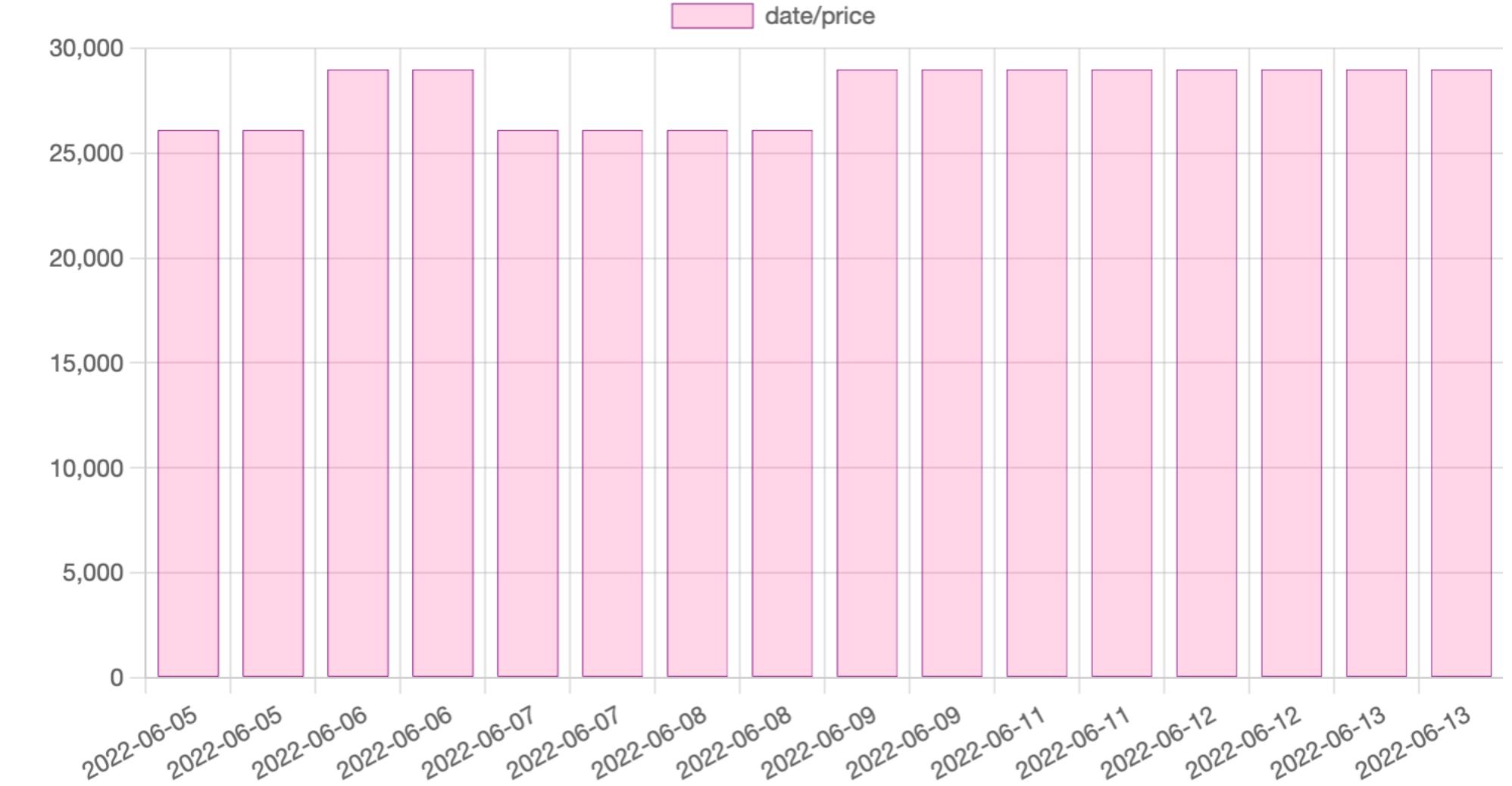
My Account



라로슈포제 미셀라 클렌징워터
터 센시티브 400ml
클렌징
라로슈포제
olive
26100 ₩

#좋아요 #클렌징워터
#순하고 #없이
#자극없이 #워터
#다 #않고
#때 #좋고

[See product](#)



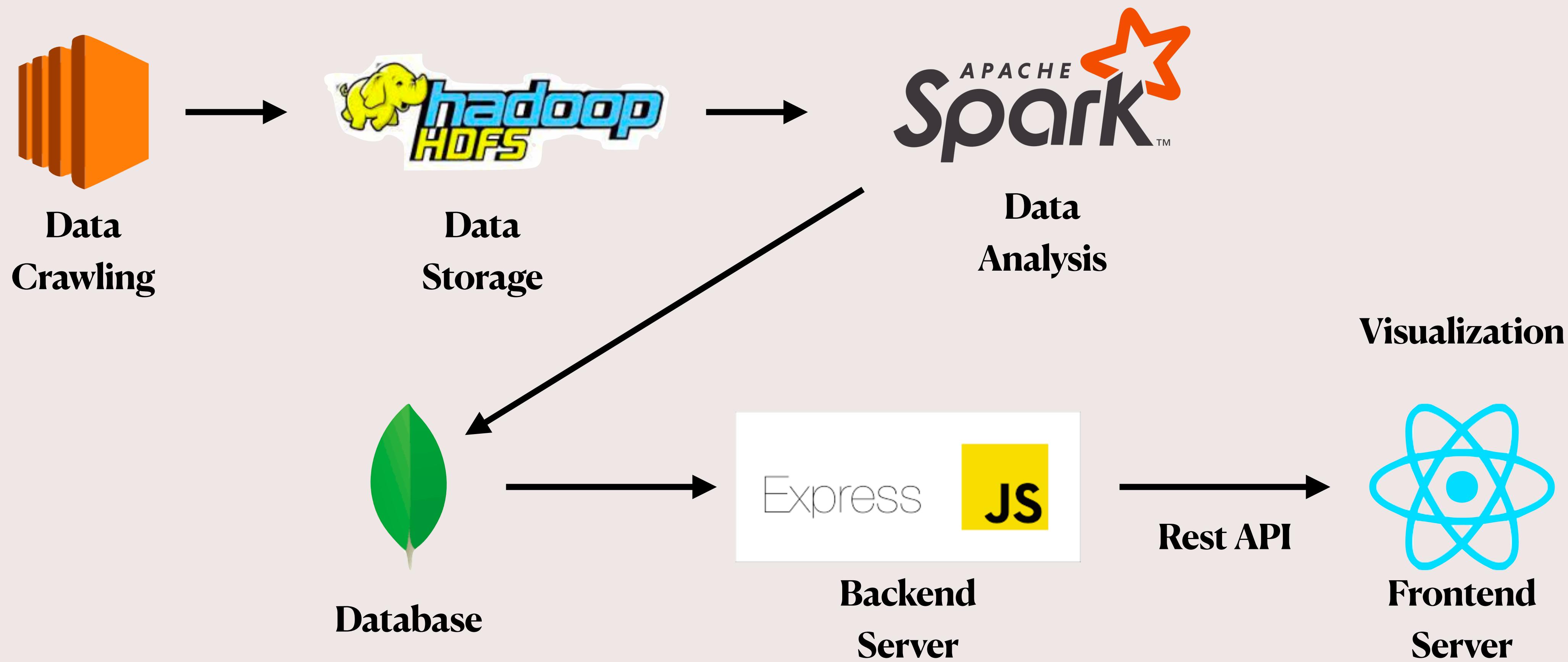
The chart displays daily price data for the product over a 9-day period. The Y-axis represents price in won, ranging from 0 to 30,000. The X-axis shows dates from June 5 to June 13, 2022. The bars are consistently pink, indicating the date and price for each day.

Date	Price (₩)
2022-06-05	26,000
2022-06-06	29,000
2022-06-07	26,000
2022-06-08	26,000
2022-06-09	29,000
2022-06-10	29,000
2022-06-11	29,000
2022-06-12	29,000
2022-06-13	29,000

Related Products

IMAGE	PRODUCT NAME	CATEGORY	BRAND	PRICE	STORE	LINK
	라로슈포제 미셀라 클렌징워터 센시티브 400ml	클렌징	라로슈포제	26100 ₩	olive	Open

Final Structure



Scrum

- Every Thursday
- Scrum Content
 - Check progress
 - Feedback
 - Set next week's goal

The image displays two side-by-side screenshots of a project management application interface.

Left Screenshot (Board view):

- Header:** Online Cosmetics Price Comparison
- Section:** Todo
- Columns:** In progress (1), Completed (12)
- In progress tasks:**
 - 웹사이트 개발 (Ahidjon Abdullaev)
- Completed tasks:**
 - Webpage wireframe (Juhee Jung)
 - hadoop upload (Juhee Jung)
 - 쿠팡 크롤링 (윤동주)
 - 무신사 크롤링 (Ahidjon Abdullaev)
 - 몰리브영 크롤링 (Juhee Jung)
 - 시각화 방안 찾아보기 (Juhee Jung, Ahidjon Abdullaev)
 - 크롤러 배포 (Juhee Jung, Ahidjon Abdullaev)
 - mongo DB 설치 (Juhee Jung)
 - comment analysis (Juhee Jung)
- Buttons:** + New

Right Screenshot (Timeline view):

- Header:** Online Cosmetics Price Comparison
- Section:** TimeLine
- Header:** Board view, Add view, No date (5), Filter, Sort, New
- Timeline:** May 2022 to June, Day 14 highlighted with a red circle.
- Tasks:**
 - 웹사이트 개발 (May 14)
 - PySpark 통해서 전처리
 - 다운로드 방법 조사
- Buttons:** + New
- Section:** 요구사항, List view
- Section:** 회의록
 - 0512 1차 스크럼 회의
 - 0519 2차 스크럼 회의
 - 0526 3차 스크럼 회의 (1 comment)
 - 0602 4차 스크럼 회의 (2 comments)
 - 0609 5차 스크럼 회의 (1 comment)
 - 0611 6차 스크럼 회의

Individual Role

Akhidjon

- 무신사 크롤러 개발 및 최적화
- 웹페이지 개발
- Backend & Frontend

윤동주

- Scrum Master
- 쿠팡 크롤러 개발
- 쿠팡 데이터 전처리

정주희

- Project Manager
- 올리브영 크롤러 & 리뷰 크롤러 개발
- 올리브영 및 무신사 데이터 전처리
- 크롤러 배포 및 Data Storage
- Cluster 구축 및 Trouble Shooting

Q&A

Thank you