

목차

1. 프로젝트 개요
2. 프로젝트 과정
3. 프로젝트 내용
4. 결론 및 기대 효과
5. 한계점 및 향후 계획

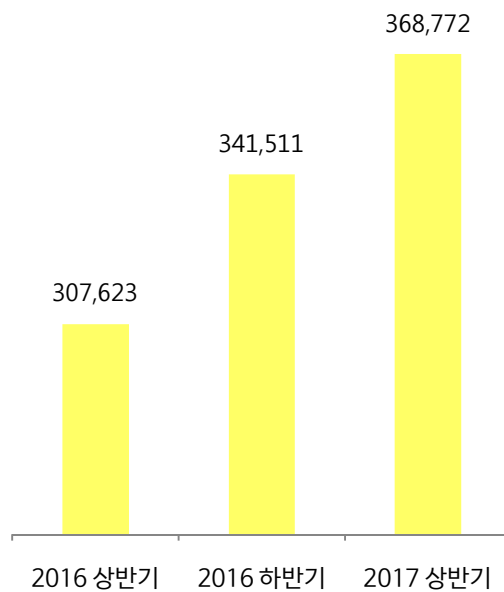
1. 프로젝트 개요

1. 프로젝트 개요

1) 개발 배경 : 국내 온라인 쇼핑몰 시장 상황

전체 시장 규모

온라인 쇼핑 매출액 현황(억원)



2017년 상반기 약 36조 8,700억원, 지속적인 성장세

온라인 쇼핑 시장 주요 이슈



네이버 쇼핑의 성장
본격 시장 진입 3년 만에 3위



인공지능
AI기술 활용한 시스템 구축

1. 프로젝트 개요

1) 개발 배경 : 스마트스토어 현 상황분석

네이버 쇼핑 플랫폼
“샵N” – “스토어팜” – “스마트스토어”



무료 온라인 창업 쇼핑몰 지원 플랫폼, 입점·판매수수료를 없애 비용 절감에 민감한 중소기업인 공략
현재 10만명의 판매자가 스마트스토어를 통해 온라인 사업을 새롭게 시작, 성장 이중 1만명 이상이 연매출 1억원을 넘어섰다.
올 3월 기준 중소기업인들의 온라인 매출 중 90%가량은 네이버의 자체 인터넷 쇼핑몰인 스마트스토어(스토어팜)를 거친 것으로 나타났다.

1. 프로젝트 개요

2) 프로젝트의 주제 및 기획의도

“네이버 스마트스토어 구매평 분석을 통한 마케팅 인사이트 도출”

— 크롤링/데이터 시각화를 바탕으로



소비자 정보의 양과 질 측면에서 중요한 “구매평”
웹크롤링, 데이터 분석 및 시각화 등을 적용
중소상공인들의 “구매자 needs” 파악에 도움

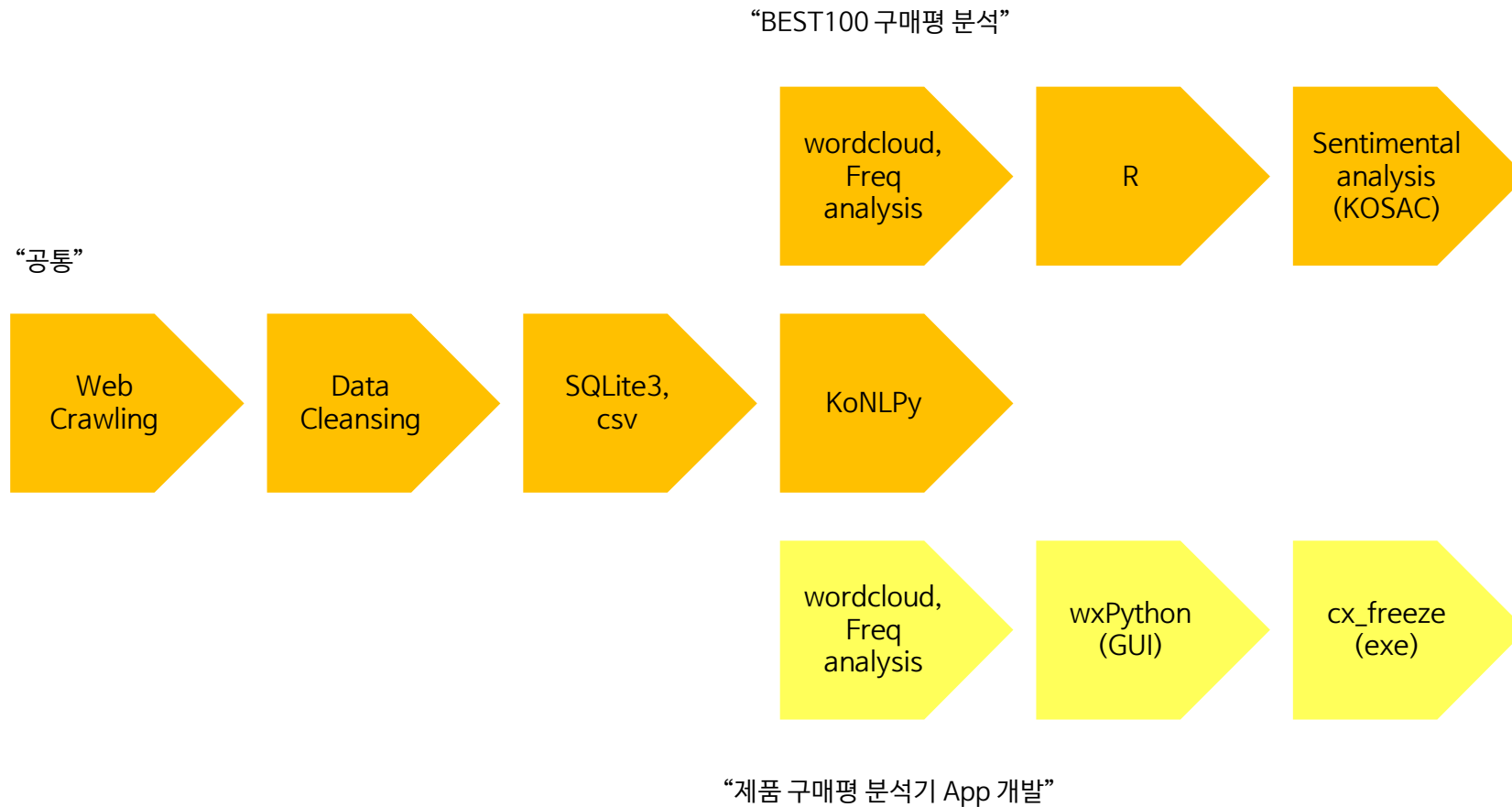
“카테고리 별 BEST100 구매평 분석”, “제품 구매평 분석기 App 개발”을 목표로 프로젝트 수행

2. 프로젝트 과정

1) 개발 과정 요약



프로젝트 세부절차



3. 프로젝트 내용 : 공통

1) 데이터 수집 : 크롤링

```
Crawling.py x Nouns.py x navie.py x
7 # 입력받은 URL을 바탕으로 json 형식의 데이터를 요청한다
8 def get_request_url(url_all):
9     try:
10         response = requests.get(url_all)
11         if int(response.status_code) == 200:
12             p = re.compile('premium.json')
13             if bool(p.search(url_all)) == True:
14                 print("PremiumComment Url Request Success")
15                 return response.text
16             else:
17                 print("GeneralComment Url Request Success")
18                 return response.text
19         except Exception as e:
20             print(e)
21             print("Error for URL")
22             return None
23
24 # 일반상품평과 프리미엄 상품평으로 나누어서 데이터를 받아온다
25 def getComment(s_name, p_number):
26     url01 = 'http://smartstore.naver.com/'
27     url03 = '/products/'
28     url05 = '/purchasereviews/premium.json?sortType=PURCHASE_REVIEW_CREATED&evaluation'
```

상세정보	구매평 2,327	Q&A 31	반품/교환정보
젤 저렴한거 같아요			
너무 저렴하게 잘샀어요 ㅎㅎ 상품도 아주 만족스럽고 5000회? 보증된다니까 물력은 안심하고 오래 쓸수있을거같아요 고장나면 하나 더 사죠 뭐 ㅎㅎ			
색상: 블랙			
ekth**** 2018.05.27 조회수 0 신고			
이빠요ㅎㅎ			
색상도 맘에들고 그림감도 나쁘지않네요 잘쓸게요 ㅎㅎ			
색상: 화이트			
id27**** 2018.05.23 조회수 7 신고			
맘에 들어요			
크다고 해서 걱정했는데 나한테는 딱 적당한 크기였음, 싼값에 산건데 기대 이상이라 만족함.			
색상: 블랙			
tng**** 2018.05.23 조회수 6 신고			
괜찮네요			
가성비 좋은 제품이라, 유망하고, 또 사용감도 좋네요, 맘에 들어요, 다른제품들도, 편하고, 잘샀네요			
색상: 화이트			
cool**** 2018.05.22 조회수 41 신고			

네이버 스토어팜 페이지에서 상품 구매자가 남긴 “구매평(프리미엄/일반) 크롤링”

3. 프로젝트 내용 : 공통

1) 데이터 수집 : 크롤링

```
← → C smartstore.naver.com/klairs/products/119945641/purchasereviews/premium.json?sortType=PURC... ☆
{
  "bSuccess" : true,
  "htReturnValue" : {
    "message" : "",
    "gridType" : "",
    "pagedResult" : {
      "content" : [{ "id": "416797742", "title": "만족", "memberId": "cha*****", "createdDate": "2018-05-12
17:32:34", "gradeText": "추천", "gradeIcon": "icon_premium_good", "contentViewUrl": "http://order.pay.naver.com/purchaseReview/416797742/premiumReview/view/shopn", "contentType": "TEXT", "best": false, "contents": null, "contentsSummary": "무난하게 좋은 토너예요. 스킨팩하기도 좋고 향도 좋아요. 가격도 비싸지 않아서 짹짹 쓰고 있어", "hitCount": 0, "reportable": true, "reportStatusCheckUrl": "http://order.pay.naver.com/purchaseReview/416797742/reportStatusCheck", "reportUrl": "http://order.pay.naver.com/purchaseReview/416797742/report", "purchasedOptionContents": null, "purchasedProductName": "클레어스 서클프레퍼레이션 페이스 토너 / 수분은 채워주고 피부결은 부드럽게", "blind": false, "resources": [], "comments": []}, {"id": "415347831", "title": "좋아요", "memberId": "joli****", "createdDate": "2018-05-09
21:35:56", "gradeText": "적극추천", "gradeIcon": "icon_premium_verygood", "contentViewUrl": "http://order.pay.naver.com/purchaseReview/415347831/premiumReview/view/shopn", "contentType": "TEXT", "best": false, "contents": null, "contentsSummary": "병풀추출물 있어서 진정효과에도 뛰어나게 좋아요~ 배송발라서 좋았어요 수고하세요~ 클레어스 변장하길!", "hitCount": 1, "reportable": true, "reportStatusCheckUrl": "http://order.pay.naver.com/purchaseReview/415347831/reportStatusCheck", "reportUrl": "http://order.pay.naver.com/purchaseReview/415347831/report", "purchasedOptionContents": null, "purchasedProductName": "클레어스 서클프레퍼레이션 페이스 토너 / 수분은 채워주고 피부결은 부드럽게", "blind": false, "resources": [], "comments": []}, {"id": "412990119", "title": "개좋음", "memberId": "hyjin****", "createdDate": "2018-05-05
12:50:24", "gradeText": "적극추천", "gradeIcon": "icon_premium_verygood", "contentViewUrl": "http://order.pay.naver.com/purchaseReview/412990119/premiumReview/view/shopn", "contentType": "TEXT", "best": false, "contents": null, "contentsSummary": "피부속속 피부 멍멍 피부 꿀꿀 원하시면 사시면 될 것 같습니다 앞으로도 오지게 맞습니다", "hitCount": 3, "reportable": true, "reportStatusCheckUrl": "http://order.pay.naver.com/purchaseReview/412990119/reportStatusCheck", "reportUrl": "http://order.pay.naver.com/purchaseReview/412990119/report", "purchasedOptionContents": null, "purchasedProductName": "클레어스 서클프레퍼레이션 페이스 토너 / 수분은 채워주고 피부결은 부드럽게", "blind": false, "resources": [], "comments": []}, {"id": "412505838", "title": "아침저녁으로 쓰기 좋네", "memberId": "gkdl****", "createdDate": "2018-05-04 14:15:47", "gradeText": "적극추천", "gradeIcon": "icon_premium_verygood", "contentViewUrl": "http://order.pay.naver.com/purchaseReview/412505838/premiumReview/view/shopn", "contentType": "TEXT", "best": false, "contents": null, "contentsSummary": "디딤터파님의 착한 토너에서 보고 싶어요 ㅎㅎ 생각보다 훨씬 가볍고 순한 느낌의 토너예요
```

만족도

“gradeText”: “추천”

댓글 내용

“contentsSummary”: 무난하게 좋은 토너예요. 스킨팩하기도 좋고 향도 좋아요. 가격도 비싸지 않아서 짹짹 쓰고 있어

날짜

“createdDate”: “2018-05-04 14:15:47”

“Json 형식”으로 데이터를 요청하여 원하는 정보를 수집

3. 프로젝트 내용 : 공통

2) 데이터 정제

```
k += ' ' + str(i)
data['content'][j] = k
j += 1

# 프리미엄 댓글과 일반 댓글의 형식을 맞추기 위해서 프리미엄의 '적극추천', '추천'
j = 0
for i in data['prefer']:
    if str(i) == '적극추천' or str(i) == '추천':
        data['prefer'][j] = '만족'
        j += 1
    elif str(i) == '추천안함':
        data['prefer'][j] = '불만'
        j += 1

return data

# 입력받은 URL 에서 스토어명과 상품번호를 정규표현식을 통해 추출한다
def main():
    url = []
    p = re.compile('http://')
    for i in count():
        url.append(str(input('{0}번째 url을 입력하세요 : '.format(i+1))))
```

프리미엄구매평 (604)

✓ 작성일순 | 조회수순

전체 ▼

전체

적극추천

추천

보통

추천안함

젤 저렴한거 같아요



적극추천

너무 저렴하게 잘샀어요 ㅎㅎ 상품도 아주 만족스럽고 5000회? 보증된다니까 클럭은 안심하고 오래 쓸수있을거같아요 고장나면 하나 더 사죠 뭐 ㅎㅎ

색상: 블랙

ekth**** | 2018.05.27 | 조회수 0 | 신고

더보기 ▼

구매평 (1723)

전체

만족

보통

불만



만족

배송빨랐어요

색상: 블랙

kajw**** | 2018.05.28 | 신고



만족

완전만족합니다. 디자인도 이쁘고, 그림감도 좋고, led가 계속 바뀌는것도 무척 예뻐요. 최고입니다

색상: 화이트

bakr**** | 2018.05.28 | 신고

만족, 보통, 불만의 “3점 척도로 통일” 시키기 위해 리코딩

3. 프로젝트 내용 : 공통

2) 데이터 정제

```
75 data = pd.DataFrame(result, columns=('type', 'prefer', 'content', 'date'))
76
77 j = 0
78 p = re.compile('\w+')
79 for item in data['content']:
80     b = item.replace('\n', '')
81     c = p.findall(b)
82     k = ''
83     for i in c:
84         k += ' ' + str(i)
85     data['content'][j] = k
86     j += 1
```

-> R에서 특수문자 중 일부가 데이터프레임을 망가뜨려 형태소 분석에 문제가 발생 : 낮은 분석 정확도

“정규표현식으로 특수문자 제거”

3. 프로젝트 내용 : 공통

3) 데이터 저장



type	prefer	content	date
0	1	저게 정말 좋아요. 가격도 저렴하고...	2023-10-27
1	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
2	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
3	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
4	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
5	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
6	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
7	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
8	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
9	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
10	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
11	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
12	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
13	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
14	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
15	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
16	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
17	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
18	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
19	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
20	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
21	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
22	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
23	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
24	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
25	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
26	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
27	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
28	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
29	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
30	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
31	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
32	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
33	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
34	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
35	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
36	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
37	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
38	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27
39	1	가격이 저렴하고, 성능도 좋네요.	2023-10-27

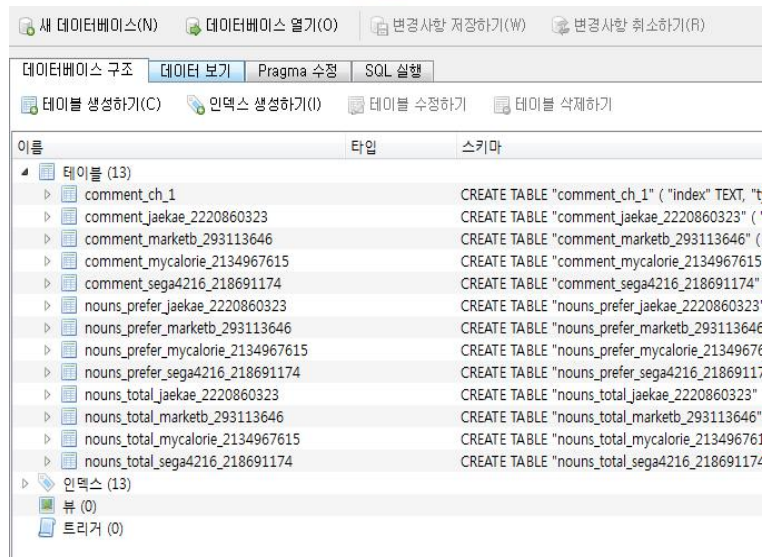
데이터 프레임
CSV로 데이터 저장
(타입, 만족도, 내용, 날짜)



명사 추출 및 감정 분석 백데이터

3. 프로젝트 내용 : 공통

3) 데이터 저장



특정 스토어팜의 제품 댓글 내용
“전체 명사 및 해당 명사의 빈도 수”
“만족도별 명사 및 해당 명사의 빈도 수”



Python 내장 DBMS, SQLite3 활용
각각의 Dataframe을 table로 저장

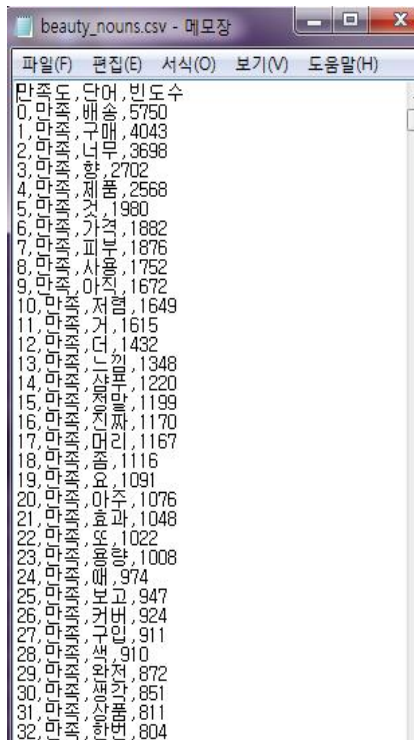
3. 프로젝트 내용 : 공통

4) 명사 추출 : KoNLPy 패키지 Twitter

```
nlp = Twitter()

# 만족,보통,불만 카테고리 나누어서 문장의 명사 빈도수를 추출한다
a = ['만족', '보통', '불만']

for i in a:
    _data = data[data['prefer'] == str(i)]
    text = ''
    for j in _data['content']:
        try:
            text += ' ' + str(j)
        except Exception as e:
            print(e)
    nouns = nlp.nouns(text)
    count = Counter(nouns)
    a = []
    b = []
    for k in count.items():
        a.append(k[0])
        b.append(k[1])
```



명사 빈도수 분석 후, 가장 많이 나온 단어부터 정렬한 데이터프레임 생성하여 csv로 저장

3. 프로젝트 내용 : BEST100 구매평 분석

5) R을 활용한 데이터 전처리 & 시각화

#csv파일 → 데이터프레임 저장 :

```
baby_nouns <- read.csv("baby_nouns.csv", header = T, stringsAsFactors = F, fileEncoding = "UTF-8")
```

##만족도별 데이터셋 분류 :

```
baby_good <- baby_nouns[baby_nouns$만족도 == "만족", ]
```

###단어컬럼에서 두 글자 이상인 행만 저장 :

```
baby_nouns_two <- filter(baby_nouns, nchar(단어) >= 2)
```

####공통빈출단어 배제하기위한 변수 리코딩 :

```
baby_good <- baby_good[baby_good$단어 != "배송" & baby_good$단어 != "너무" & baby_good$단어 != "가격" & baby_good$단어 != "구매" & baby_good$단어 != "진짜" & baby_good$단어 != "아직" & baby_good$단어 != "정말" & baby_good$단어 != "매우" & baby_good$단어 != "생각" & baby_good$단어 != "제품" & baby_good$단어 != "상품" & baby_good$단어 != "아주" & baby_good$단어 != "완전", ]
```


3. 프로젝트 내용 : BEST100 구매평 분석

5) R을 활용한 데이터 전처리 & 시각화



“빈도분석 막대그래프(ggplot2)”

```
top30 <- comment_nouns_two %>% arrange  
(desc(빈도수합계)) %>% # 빈도수 상위30개 내림차순 정렬  
  
order <- arrange(top30, 빈도수합계)$단어  
# 빈도수 상위30개 저장  
  
ggplot(data=top30, aes(x=단어, y=빈도수합계, fill=단어)) +  
  ylim(0,50000) +  
  geom_col() +  
  coord_flip() +  
  scale_x_discrete(limit = order) +  
  geom_text(aes(label=빈도수합계), hjust=-0.3)
```



“워드클라우드(wordcloud)”

```
pal <- brewer.pal(9,"Set1")  
# 만족:Set1, 보통:Blues, 불만:OrRd # 색깔변수 지정  
set.seed(1234) # 난수고정  
  
wordcloud(words = baby_good$단어,  
  freq = baby_good$빈도수,  
  max.word = 100,  
  random.order = F,  
  random.color = T,  
  rot.per = .1,  
  scale = c(5, 0.5),  
  colors = pal,  
  )
```

3. 프로젝트 내용 : BEST100 구매평 분석

6) 감정분석 - KOSAC 사전을 이용한 학습 데이터 생성

Home
Corpus
Lexicon
Publications

Korean Sentiment Lexicon Release Page

The sentiment lexicon can be used solely for academic and research purposes.

o [Agreement Form](#) Download

Send us the agreement to kosac.snu@gmail.com

Sentiment Lexicon

Version 1.0

o [README](#)

o [Annotation Scheme \(PACLIC26\)](#)

o [Download](#)

서울대학교 컴퓨터 언어학과에서 만든 “KOSAC 사전”을 이용하여 긍정 (POS), 부정 (NEG), 중립 (NEUT)의 단어를 학습 데이터로 생성

8622	시대/NNG;정신/NNG	NEG	1
8623	시대/NNG;정신/NNG;예/JKB	NEG	1
8624	시도/NNG	POS	1
8625	시도/NNG;하/XSV	POS	1
8626	시련/NNG	NEG	1
8627	시멘트/NNG	NEG	1
8628	시멘트/NNG;길/NNG	NEG	1
8629	시멘트/NNG;덩어리/NNG	NEG	1
8630	시사/NNG	NEUT	1
8631	시사/NNG;하/XSV	NEUT	1
8632	시선/NNG	POS	1
8633	시선/NNG;이/JKS	POS	1
8634	시선/NNG;이/JKS;두드리지/	POS	1
8635	시설/NNG	POS	1
8636	시설/NNG;을/JKO	POS	1
8637	시설/NNG;을/JKO;갖추/VV	POS	1
8638	시약/NNG	NEG	1
8639	시약/NNG;을/JKO	NEG	1
8640	시인/NNG	NEUT	1
8641	시인/NNG;의/JKG	NEUT	1

3. 프로젝트 내용 : BEST100 구매평 분석

6) 감정분석 - NaiveBayseClassifier 를 이용한 데이터 학습 및 분석

```
Crawling.py x Nouns.py x navie.py x
25
26 # 중복 단어를 제거하기 위해서 set로 만들었다가 list로 변환시킨다
27 positive_vocab = list(set(pos_dic)) #총 1830개 단어
28 negative_vocab = list(set(neg_dic)) #총 1623개 단어
29 neutral_vocab = list(set(neu_dic)) #총 340개 단어
30
31 def word_feats(words):
32     return dict((word, True) for word in words)
33
34 # 사전의 긍정, 부정, 중립단어를 navie bayes에 학습시킬 준비를 한다
35 positive_features = [(word_feats(pos), 'pos') for pos in positive_vocab]
36 negative_features = [(word_feats(neg), 'neg') for neg in negative_vocab]
37 neutral_features = [(word_feats(neu), 'neu') for neu in neutral_vocab]
38
39 # 트레인 데이터셋 생성 완료! naive bayes에 학습 시킨다
40 train_set = negative_features + positive_features + neutral_features
41 classifier = NaiveBayesClassifier.train(train_set)
42
43 # 예측하기 위해 데이터셋 준비(_data.csv 앞에 글자만 작성, beauty를 예시로 써둠)
44 name = "beauty"
45
46 df = pd.read_csv("C:/naverstore/final/%s_data.csv"%(name))
47 data = df[df['content'].notnull()]
```

KOSAC의 긍정, 부정, 중립 단어 SET를 “NaiveBayseClassifier를 통해 학습”시킨 후 네이버 댓글을 분석

3. 프로젝트 내용 : BEST100 구매평 분석

6) 감정분석 - KOSAC 사전 수정

긍정적인 문장

w_count_pos - 메모장

파일(F) 편집(E) 서식(O) 보기(V)

sentence, count
101, 만족합니다, 1969
22, 좋아요, 604
327, 감사합니다, 247
197, 좋아요좋아요, 187
562, 좋습니다, 161
134, 맘에들어요, 154
256, 아주좋아요, 118
165, 좋아요 좋아요, 109
186, 빠른배송 감사합니다, 104
140, 배송빠르고 좋아요, 87
79, 빠른배송감사합니다, 77
466, 배송도 빠르고 좋아요, 74
89, 아주 좋아요, 68
330, 빠른 배송 감사합니다, 68
208, 배송 빠르고 좋아요, 66
504, 좋아요좋아, 63
109, 좋아요 * *, 62
265, 이빠요, 60
324, 좋아요 * *, 59
2175, 마기가 좋아해요, 50
74, 맘에 들어요, 43
181, 마음에 들어요, 37
104, 배송빠르고 좋아요, 36
333, 만족합니다, 34
563, 예빠요, 33
183, 마음에들어요, 31
325, 매우 만족합니다, 30

중립적인 문장

w_count_neu - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

sentence, count
339, 괜찮네요, 19
14951, 할인받고 저렴하게 구매했어요 동이라
2776, 너무맘에들어요, 9
8019, 괜찮은거 같아요, 7
3097, 괜찮은것 같아요, 7
1703, 늘 쓰던거라 만족합니다, 6
2628, 좋은것 같아요, 6
940, 색깔도 이쁘고 촉감도 좋은듯, 6
2870, 너무이빠용, 6
16209, 늘쓰는제품, 5
326, 맘에들어요, 5
10356, 조아여궁긔, 4
2769, 괜찮은듯해요, 4
15000, 너무귀엽고 예빠요 *, 4
1718, 늘쓰던제품, 4
239, 배송빠르고 질도 좋아요 맘에 듭니다, 4
1726, 사용전 다른브랜드에비해 저렴해서 좋은
1270, 너무 맘에 들어요, 4
14649, 마기가 기어다니기 시작해서 주문했는
390, 넘나 귀엽구 배송도 좋아요, 4
14483, 애들이 너무좋아하는거 같아요 안전한
14505, 마기가 기어다니기 시작해서 주문했는
969, 부드럽고 너무 맘에 듭니다, 4
2777, 너무 이빠용, 4
602, 배송 빨라요 질이 마음에 들어요, 4
603, 배송 빨라요 질이 너무 마음에 들어요, 4
14620, 더구매해마졌어요, 3

부정적인 문장

w_count_neg - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

sentence, count
43, 잘받았습니다, 404
29, 잘받았어요, 255
19, 배송빨라요, 208
132, 잘 받았습니, 156
30, 너무좋아요, 104
37, 잘 받았어요, 74
32, 너무 좋아요, 72
60, 만족해요, 54
268, 조아요조아요, 51
452, 배송 빨라요, 49
41, 저렴하고 좋아요, 45
122, 마기가 좋아해요, 45
591, 괜찮아요, 42
63, 배송빠르네요, 39
18, 귀여워요, 38
51, 조아요조아, 36
2207, 좋아요 잘쓸게요, 32
4210, 저렴하고좋아요, 31
345, 잘받았어요, 30
901, 잘쓰고있어요, 29
203, 잘쓰겠습니다, 29
247, 이빠네요, 27
798, 잘쓸게요, 26
449, 좋네요, 26
552, 조아요, 25
4386, 잘쓸게요 * *, 25
47, 아주조아요, 24

[부정확한 분석 결과]

: ‘잘받았습니다’, ‘잘받았어요’, ‘배송빨라요’ 등을 부정적인 단어로 인식

KOSAC 사전에 ‘잘’, ‘잘받다’ ‘배송빨라’ 등을 긍정의 단어로 추가 하여 “재학습”

3. 프로젝트 내용 : BEST100 구매평 분석

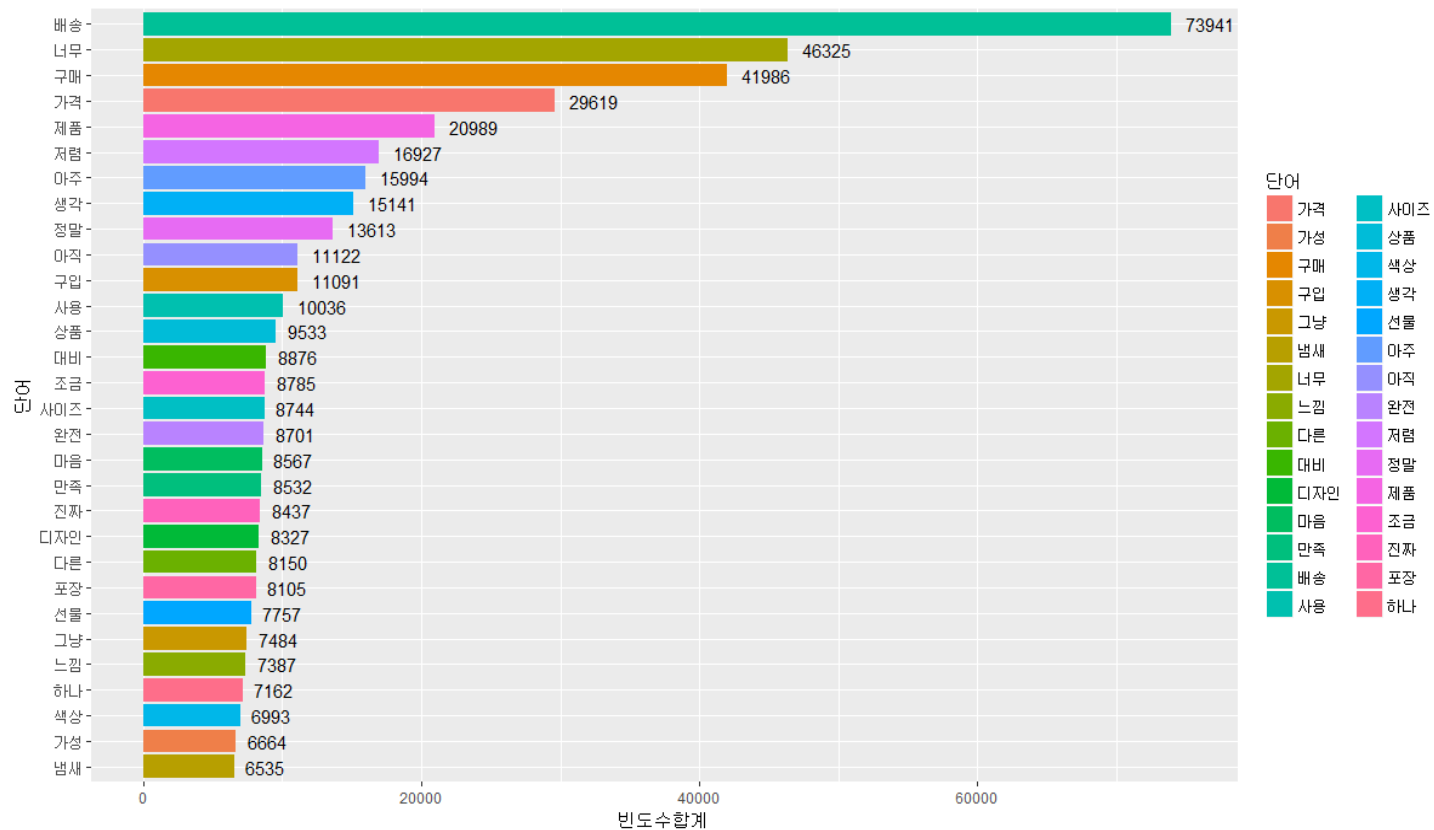
6) 감정분석 - 최종 결과

digital			fashion_a			fashion_c		
긍정 66%	중립 33%	부정 1%	긍정 60%	중립 39%	부정 1%	긍정 62%	중립 36%	부정 2%
만족합니다,1587	보통입니다,14	이쿠조쿠요,3	만족합니다,831	보통입니다,13	무난하네용,3	만족합니다,585	보통입니다,10	사진 그대로입니다,3
잘받았습니다,449	괜찮은거같아요,13	그립감 좋네요,3	좋아요,244	그냥저냥 괜찮아요,6	머 가격대로 입니다,3	좋아요,139	보통이에요,5	강추드립니다,3
감사합니다,427	좋은것같아요,7	아직안써봄,3	잘받았습니다,191	그냥그래여,6	사진과 똑같아요,3	잘받았습니다,118	입을만해요,5	팔왜케 짧아요,2
좋아요,378	괜찮은것 같아요,7	감사합니다,2	감사합니다,113	심플하고 예뻐요,6	별로입니다,2	잘받았어요,98	괜찮아용,4	속바지로 짱,2
좋습니다,256	가성비 갑,6	너무 약함,2	잘받았어요,113	다들 이쁘다고 그래요,6	그냥저냥 ㅋㅋㅋㅋ,2	이뻐요,91	괜찮은거 같아요,4	그냥그냥그냥,2
잘 받았습니다,252	가성비갑입니다,6	잘 들립니다,2	예뻐요,107	심플하니 이뻐요,4	딱 그가격입니다,2	맘에들어요,83	그럭저럭 괜찮아요,3	생각보다 두꺼워요,2
아주좋아요,179	가성비 짱입니다,5	별로입니다,2	맘에들어요,93	괜찮은거 같아요,4	그냥저냥,2	예뻐요,56	재구매했어요,3	냉장고바지같아요,2
빠른배송 감사합니다,179	가성비갑이네요,4	굳입니다,2	이뻐요,82	좋아요 다만 사이즈 선	조아요오용,2	괜찮아요,47	믿고사는 마약잠옷,3	조아요오용,2
잘받았어요,137	좋은것 같아요,4	구우우우울,2	잘 받았습니다,70	매번 편의점에서 구매하	그냥그저그럼,2	감사합니다,46	그냥그래여,3	그냥그냥,2
빠른 배송 감사합니다,137	좋은 것 같아요,3	기대이상입니다,2	좋습니다,64	심플하니예뻐요,4	편할꺼 같아요,2	좋아요좋아요,44	그냥 괜찮아요,3	그냥 기본티,2
배송빨라요,120	괜찮은 것 같습니다,3	그립감 좋아요,2	배송빨라요,54	괜찮은거같아요,4	원터치라 착용하기 편하	배송빨라요,43	편하게 입을것 같아요,3	시원한 소재입니다,2
아주 좋아요,101	굿뜨,3	삼각대 개꿀띠,2	맘에 들어요,52	배송이 굉장히 빨라요	사진그대로입니다,1	잘 받았습니다,39	안녕 하세요 잘받았습니	시원할꺼같아요,1
빠른배송감사합니다,94	와이파이 짱짱합니다,3	써보니까 좋네요,2	편하고 좋아요,49	존존하고 면이 독특해요	오스드스드스드스드스드	그냥그래요,38	11개월아기 딱맞을것같	사진과동일,1

사전 수정 후 분류 정확도가 매우 향상 되었으나
[맞춤법이 틀린 경우, 부정적인 단어로 인식]

3. 프로젝트 내용 : BEST100 구매평 분석결과

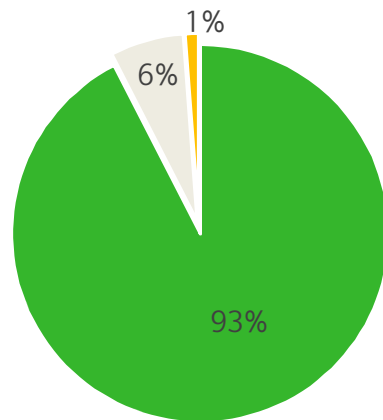
1) 네이버 스토어팜 BEST100 구매평 381,287건 (두 자 이상의 단어) - 빈도분석



3. 프로젝트 내용 : BEST100 구매평 분석결과

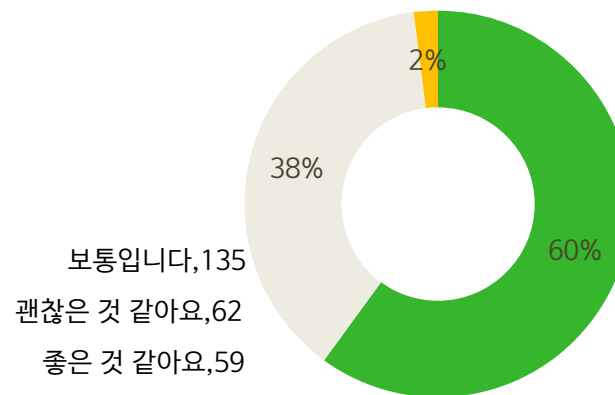
1) 네이버 스토어팜 BEST100 구매평 381,287건 (두 자 이상의 단어) - 빈도/감정분석

#전체 구매평 만족도



■ 만족 ■ 보통 ■ 불만

맛있어요,22
아직 안써봤어요,17
조아요오옹,13



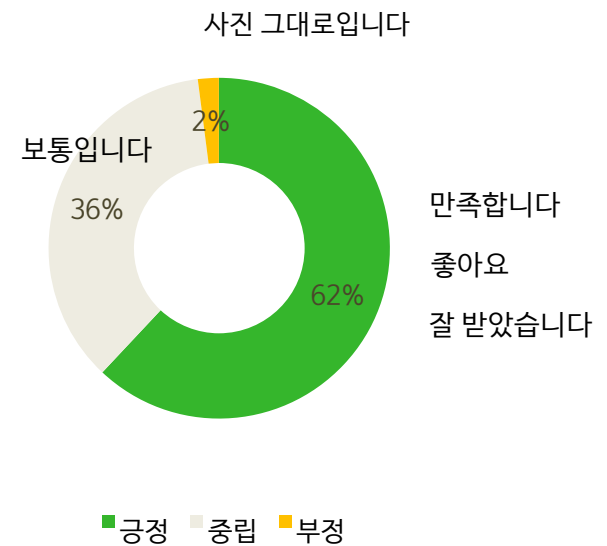
■ 긍정 ■ 중립 ■ 부정

만족합니다,13281
잘 받았습니,3579
보통입니다,135

3. 프로젝트 내용 : BEST100 구매평 분석결과

2) 패션의류 : 빈도/감정분석(18,798건)

#만족척도에 따른 워드클라우드

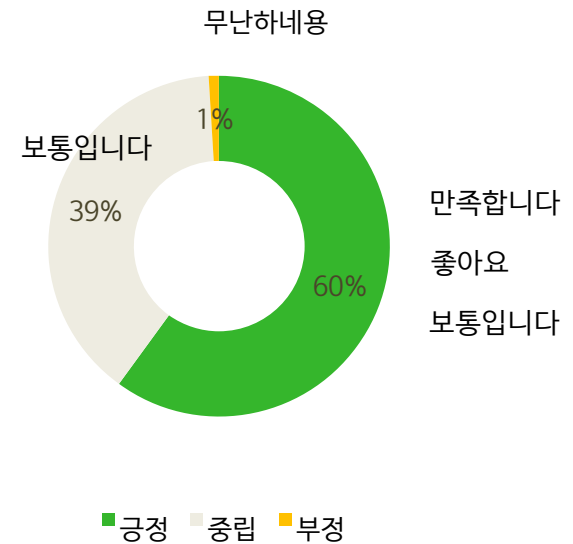


[인기상품은 “여름의상”, 주요 요인은 “색상”, “재질”, “사이즈”]

3. 프로젝트 내용 : BEST100 구매평 분석결과

3) 패션잡화 : 빈도/감정분석(30,369건)

#만족척도에 따른 워드클라우드



[인기상품은 “가방,” “신발,” “귀걸이” 등 선물, 주요 요인은 “사이즈,” “가격,” “냄새”]

3. 프로젝트 내용 : BEST100 구매평 분석결과

4) 화장품 및 미용 : 빈도/감정분석(36,526건)

#만족척도에 따른 워드클라우드



〈만족〉

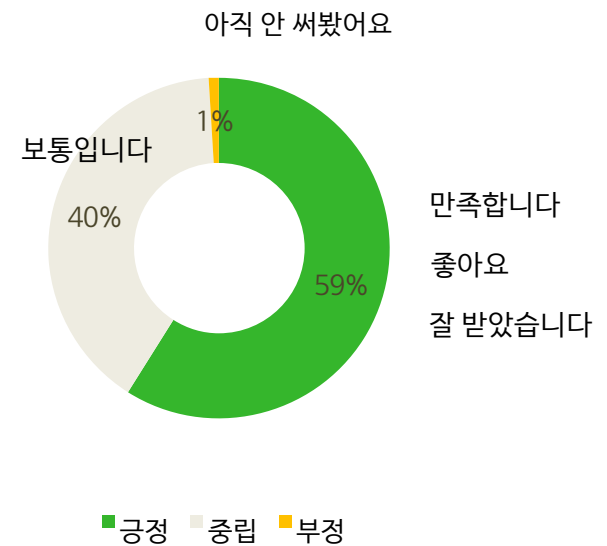


〈보통〉



〈불만〉

[인기상품은 “스킨케어”, “메이크업 제품”, 주요 요인은 ”가격”, “성능”]



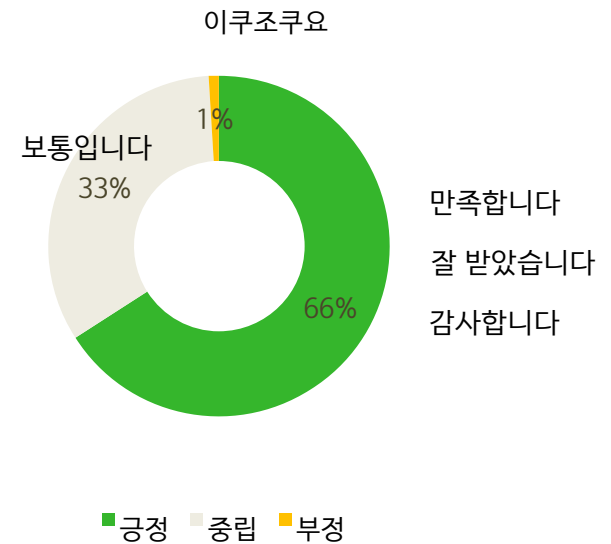
3. 프로젝트 내용 : BEST100 구매평 분석결과

5) 디지털 및 가전 : 빈도/감정분석(44,854건)

#만족척도에 따른 워드클라우드



[인기상품은 “공유기”, ”휴대폰 케이스”, ”스피커”, 주요 요인은 ”가격”, ”디자인”, ”성능”]



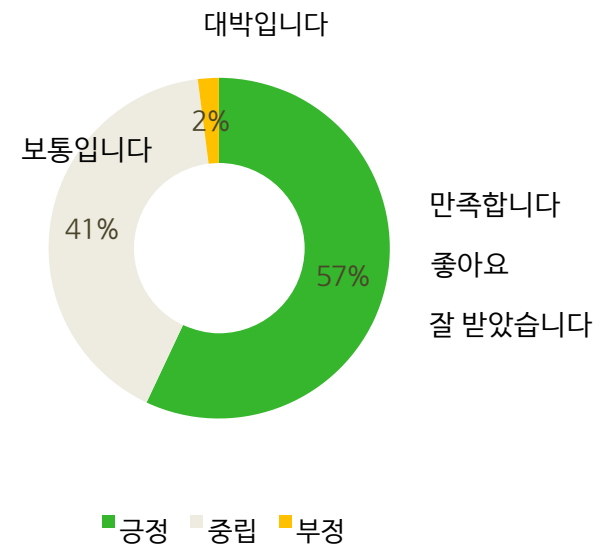
3. 프로젝트 내용 : BEST100 구매평 분석결과

6) 가구 및 인테리어 : 빈도/감정분석(57,304건)

#만족척도에 따른 워드클라우드



[인기상품은 “조립식 가구”, ”의자”, 주요 요인은 ”색상”, “가격”, ”냄새”]



3. 프로젝트 내용 : BEST100 구매평 분석결과

7) 출산 및 육아 : 빈도/감정분석(44,661건)

#만족척도에 따른 워드클라우드



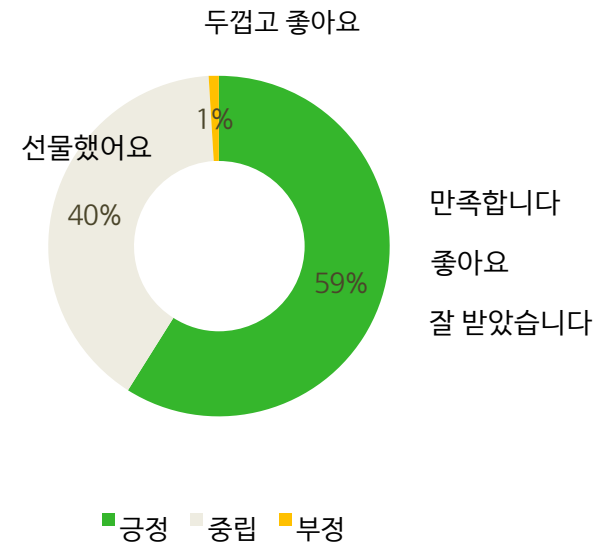
<만족>



<보통>



<불만>



[인기상품은 “아기용 멀티슈,” “이유식,” “유모차”, 주요 요인은 ”편리함”, “가격”]

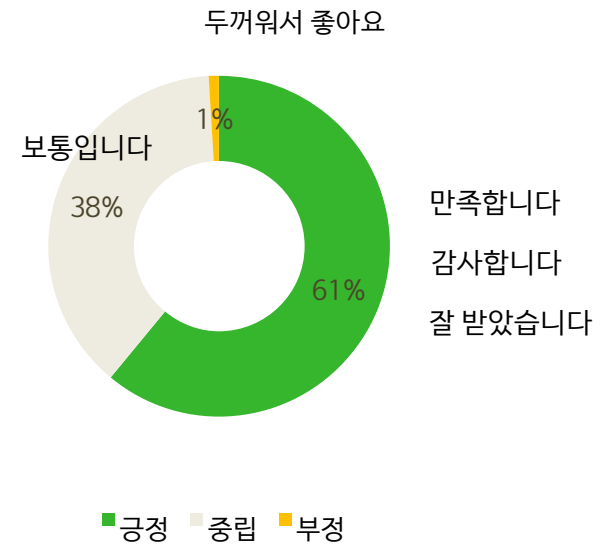
3. 프로젝트 내용 : BEST100 구매평 분석결과

8) 스포츠 및 레저 : 빈도/감정분석(33,442건)

#만족척도에 따른 워드클라우드



[인기상품은 “운동매트”, ”밴드”, 주요 요인은 ”가격”, ”사이즈”, ”성능”]



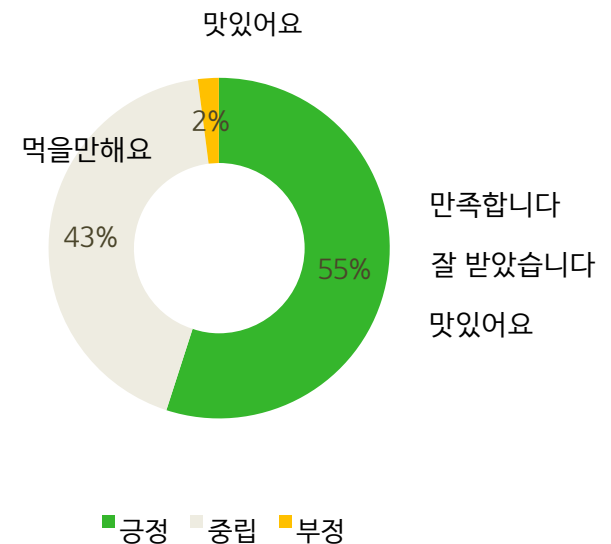
3. 프로젝트 내용 : BEST100 구매평 분석결과

9) 식품 : 빈도/감정분석(53,012건)

#만족척도에 따른 워드클라우드



[인기상품은 “선물용”, “다이어트 식품”, 주요 요인은 ”가격”, “포장”, ”식감”]



3. 프로젝트 내용 : BEST100 구매평 분석결과

10) 생활용품 : 빈도/감정분석(62,231건)

#만족척도에 따른 워드클라우드



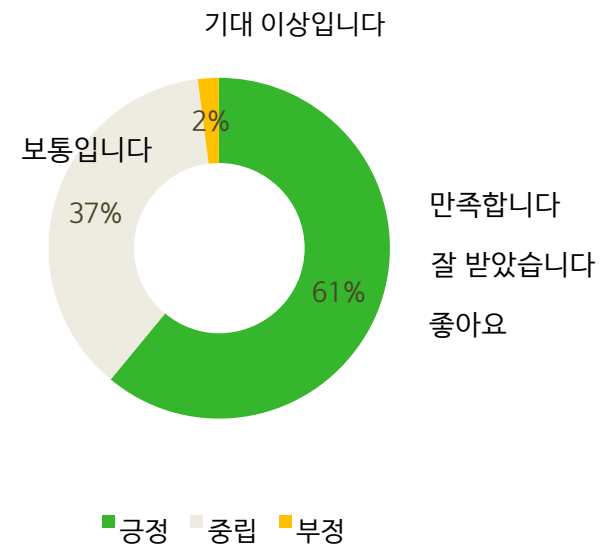
〈만족〉



〈보통〉

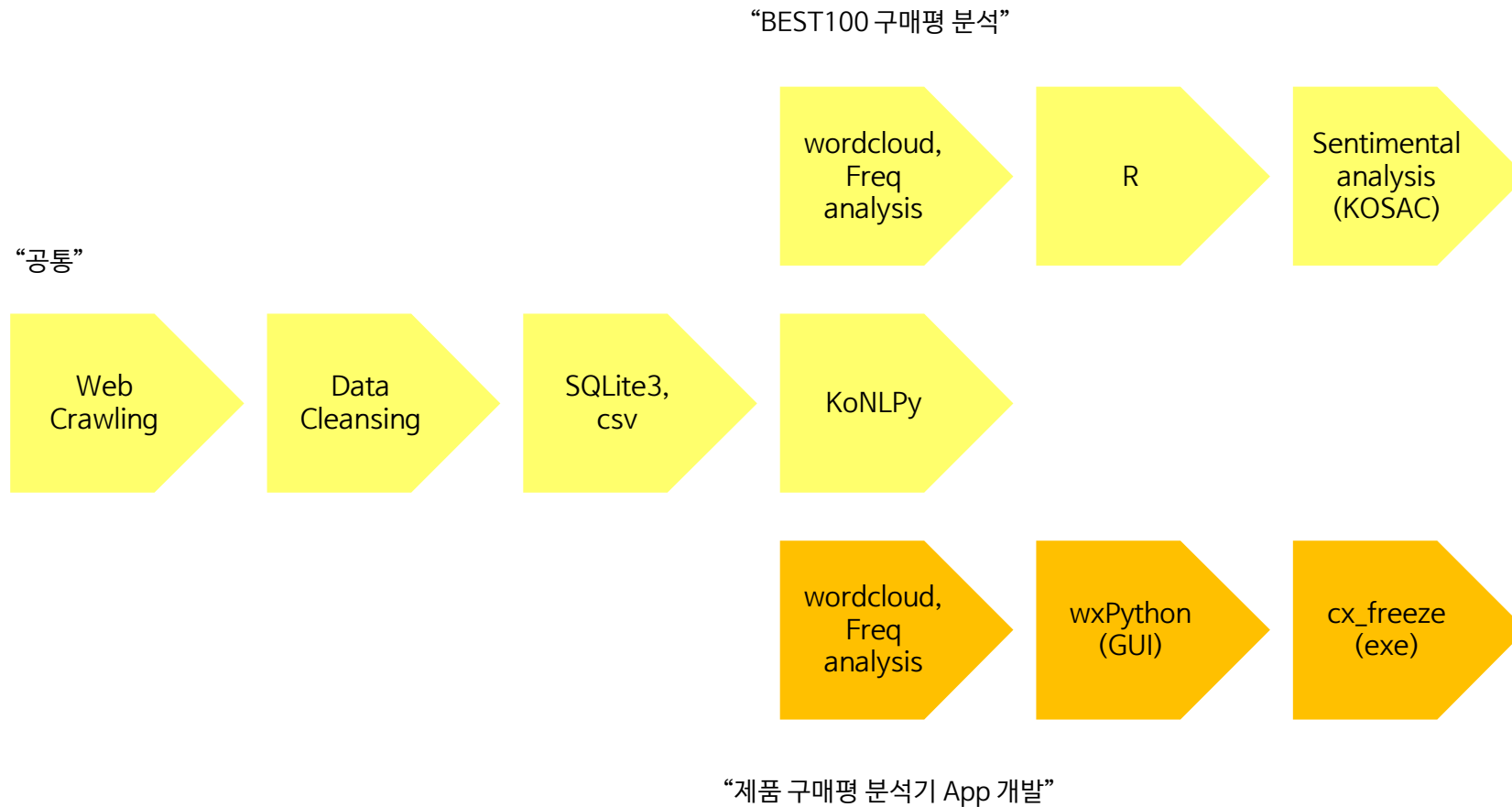


〈불만〉



[인기상품은 “충전기,” 애견용품, 주요 요인은 ”가격,” “디자인,” 성능]

프로젝트 세부절차



3. 프로젝트 내용 : 제품 구매평 분석기 App 개발

1) 제품 구매평 분석기 App 개발



“GUI기반의 프로그램 개발”

3. 프로젝트 내용 : 제품 구매평 분석기 App 개발

2) 빈도 분석 및 워드클라우드

```
#wordcloud와 barplot을 생성하여 이미지 저장
def createChart(df_list, s_name, p_number):
    count = 1

    for i in df_list:
        nouns = list(i['단어'].loc[0:9])
        nouns.reverse()
        freq = list(i['빈도수'].loc[0:9])
        freq.reverse()

        plt.figure(figsize=(5.5, 4))
        rects = plt.barh(np.arange(10), freq, align='center')
        plt.yticks(np.arange(10), nouns)

        for k, rect in enumerate(rects):
            plt.text(0.99 * rect.get_width(), rect.get_y() +
                     str(freq[k]), ha='right', va='center')

        plt.xlabel('빈도수')
        plt.title('TOP 10 단어 빈도수', size=20)

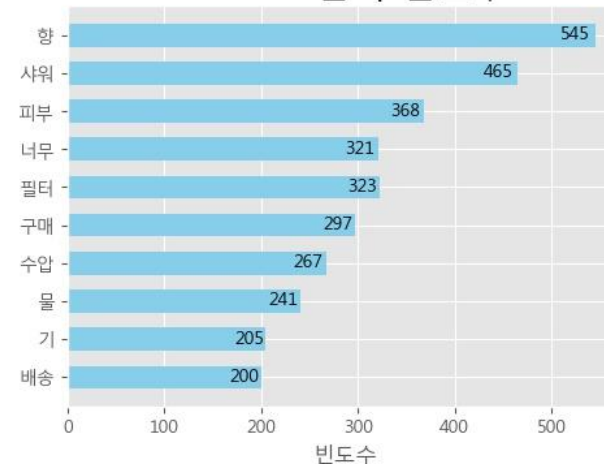
        if count == 1:
            fig = plt.gcf()
            fig.savefig('total_bar_{0}_{1}.jpg'.format(s_name, p_number))
        elif count == 2:
            fig = plt.gcf()
            fig.savefig('good_bar_{0}_{1}.jpg'.format(s_name, p_number))
        elif count == 3:
            fig = plt.gcf()
            fig.savefig('moderate_bar_{0}_{1}.jpg'.format(s_name, p_number))
        else:
            fig = plt.gcf()
```

〈Wordcloud, PIL 패키지〉



〈Matplotlib.pyplot 패키지〉

TOP 10 단어 빈도수



3. 프로젝트 내용 : 제품 구매평 분석기 App 개발

3) Py파일 모듈화 및 연결

```
manyproduct = {'klairs': '119945641'}

for oneproduct in manyproduct_:
    s_name = oneproduct
    p_number = manyproduct[s_name]

    url01 = 'http://smartstore.naver.com/'
    url03 = '/products/'
    url05 = '/purchasereviews/premium.json?sortType=PURCHASE_RE
    url_all = url01 + s_name + url03 + p_number + url05

    # print(url_all)
    source_code = requests.get(url_all)
    # print(source_code.status_code)
    plain_text = source_code.text
    mydatas = json.loads(plain_text)
    # print(mydatas)

    list_comment = []
    for comment in mydatas['htReturnValue']['pagedResult']['con
        list_comment.append('프리미엄댓글')
        list_comment.append(comment['gradeText'])
        list_comment.append(comment['contentsSummary'])
        list_comment.append(comment['createdDate'])

    result = []
    _result = []
    for item in list_comment:
        # print(item)
        result.append(item)
```



```
def get_request_url(url_all):
    try:
        response = requests.get(url_all)
        if int(response.status_code) == 200:
            p = re.compile('premium.json')
            if bool(p.search(url_all)) == True:
                print("PremiumComment Url Request Success")
                return response.text
            else:
                print("GeneralComment Url Request Success")
                return response.text
    except Exception as e:
        print(e)
        print("Error for URL")
        return None

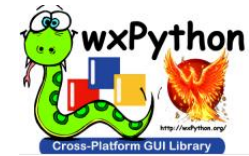
def getComment(s_name, p_number):
    url01 = 'http://smartstore.naver.com/'
    url03 = '/products/'
    url05 = '/purchasereviews/premium.json?sortType=PURCHAS
    url_all1 = url01 + s_name + url03 + p_number + url05

    url01 = 'http://smartstore.naver.com/'
    url03 = '/products/'
    url05 = '/purchasereviews/general.json?satisfactionGrad
    url_all2 = url01 + s_name + url03 + p_number + url05

    plain_text1 = get_request_url(url_all1)
    plain_text2 = get_request_url(url_all2)
```

3. 프로젝트 내용 : 제품 구매평 분석기 App 개발

4) 시각화(GUI)



mainGUI.py : 메인 화면(덧글)

```
import wx
import wx.xrc
import sqlite3
import crawlingFinal
import nounExtract

class MyFrame1(wx.Frame):
    def __init__(self, parent):
        wx.Frame.__init__(self, parent, id=wx.ID_ANY, title='네이버 스토어팜 소비자 반응 워드클라우드 && 단어 빈도 차트',
            size=wx.Size(1300, 700), style=wx.DEFAULT_FRAME_STYLE | wx.RESIZE_BORDER)

        self.SetSizeHintsSz(wx.DefaultSize, wx.DefaultSize)

        bSizer14 = wx.BoxSizer(wx.VERTICAL)

        self.m_panel14 = wx.Panel(self, wx.ID_ANY, wx.DefaultPosition, wx.DefaultSize, wx.BORDER_NONE)
        bSizer15 = wx.BoxSizer(wx.HORIZONTAL)

        self.m_staticText10 = wx.StaticText(self.m_panel14, wx.ID_ANY, u"URL :", wx.DefaultPosition, wx.DefaultSize, wx.DefaultFont,
            wx.DefaultStyle, wx.DefaultColour, wx.DefaultBackgroundColour, wx.DefaultBorder, wx.DefaultMargin, 0, wx.Wrap(-1))
        bSizer15.Add(self.m_staticText10, 0, wx.ALL, 5)

        self.txtUrl = wx.TextCtrl(self.m_panel14, wx.ID_ANY, wx.EmptyString, wx.DefaultPosition, wx.DefaultSize, wx.DefaultFont,
            wx.DefaultStyle, wx.DefaultColour, wx.DefaultBackgroundColour, wx.DefaultBorder, wx.DefaultMargin, 0, wx.Wrap(-1))
        bSizer15.Add(self.txtUrl, 1, wx.ALL, 5)
```

chartGUI : 차트 화면(워드클라우드, 빈도 차트)

```
# -*- coding: utf-8 -*-
import wx
import wx.xrc

class MyDialog(wx.Dialog):
    def __init__(self, parent, id, title, s_name, p_number):
        wx.Dialog.__init__(self, parent, id, title, size=(1200, 700))

        text = wx.StaticText(self, -1, '스토어팜 소비자 반응 워드클라우드 && 단어 빈도 차트', (20, 20))
        text.SetFont(wx.Font(26, 72, 90, 90, False, "HY수평선평"))

        panel = wx.Panel(self, -1, (20, 150), (550, 400), style=wx.SUNKEN_BORDER)
        self.picture = wx.StaticBitmap(panel)
        panel.SetBackgroundColour(wx.WHITE)

        panel2 = wx.Panel(self, -1, (600, 150), (550, 400), style=wx.SUNKEN_BORDER)
        self.picture2 = wx.StaticBitmap(panel2)
        panel2.SetBackgroundColour(wx.WHITE)

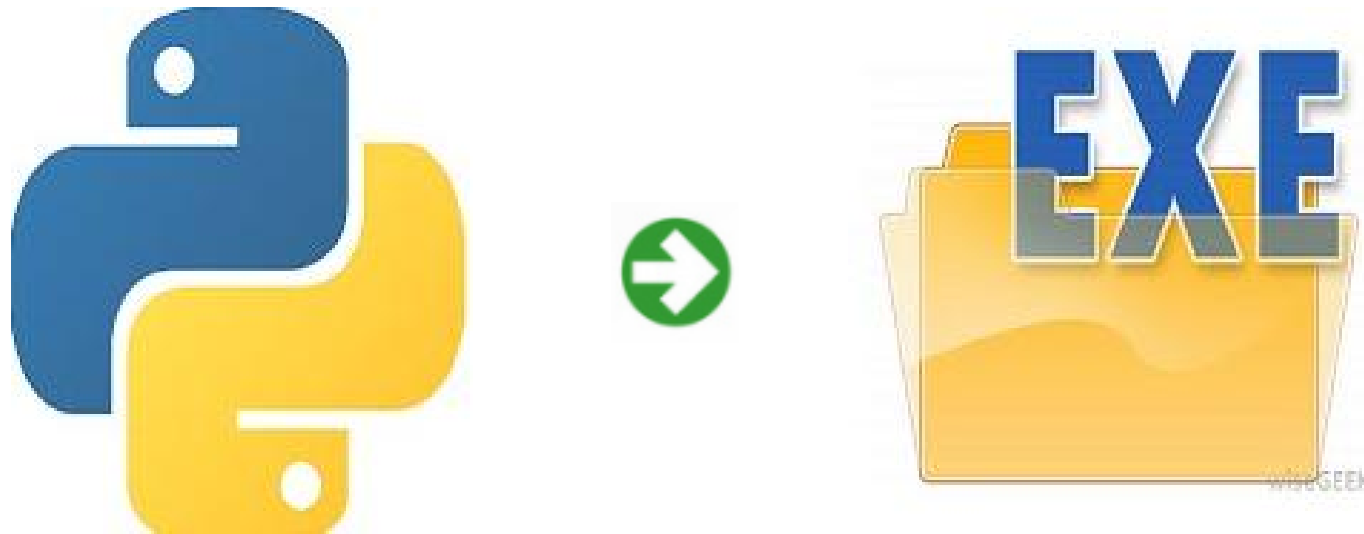
        #이미지 loads
        self.images = ['image_data/total_wc_{0}_{1}.jpg'.format(s_name, p_number), 'image_data/good_wc_{0}_{1}.jpg'.format(s_name, p_number)]
        self.images2 = ['image_data/total_bar_{0}_{1}.jpg'.format(s_name, p_number), 'image_data/good_bar_{0}_{1}.jpg'.format(s_name, p_number)]
        authors = ['전체', '만족', '보통', '불만']

        wx.ComboBox(self, -1, pos=(20, 80), size=(100, -1), choices=authors, style=wx.CB_READONLY)
        wx.Button(self, 1, 'Close', (1080, 10))
```


3. 프로젝트 내용 : 제품 구매평 분석기 App 개발

5) exe 실행파일 배포

〈cx_freeze 패키지〉



3. 프로젝트 내용 : 제품 구매평 분석기 App 개발

6) exe 실행파일 생성 및 확인

setup.py : 빌드에 관한 설정

```
from cx_Freeze import setup, Executable
import sys

buildOptions = dict(packages=["wordcloud", "idna", "re", "sys", "os", "jpy", "konpy", "m
                        excludes=["tkinter", "PyQt4.QtSql"],
                        zip_includes=[(r'C:\CommentAnalysis\venv\Lib\site-packages\wordcloud
                                     (r'C:\CommentAnalysis\venv\Lib\site-packages\wordcloud

)

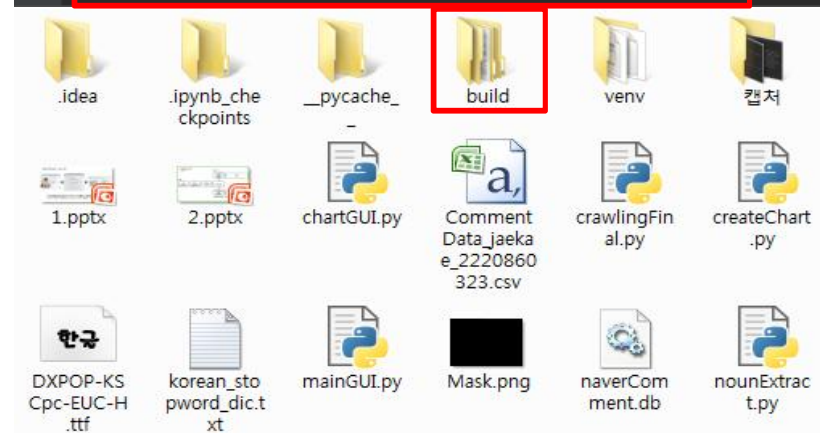
base = None
if sys.platform == "win32":
    base = "Win32GUI"

exe = [Executable("mainGUI.py", base=base)]

setup(
    name='CommentAnalyzer',
    version='0.1',
    author="IML",
    description="I'M IML!",
    options=dict(build_exe=buildOptions),
    executables=exe
)
```

실행파일 빌드

```
Terminal
+ elib\Solarize_Light2.mplstyle
✗ copying C:\CommentAnalysis\venv\lib\site-packages\matplotlib\stylelib\tableau-colorblind10.mplstyle
copying C:\CommentAnalysis\venv\lib\site-packages\matplotlib\classic_test.mplstyle
copying C:\Anaconda3\DLLs\sqlite3.dll -> build\exe.win32\sqlite3.dll
copying C:\CommentAnalysis\venv\Scripts\api-ms-win-crt-...
(venv) C:\CommentAnalysis>python setup.py build
```



4. 결론 및 기대효과

공통발생단어 배제 후,
카테고리별 or 만족도별
“차별화된 워드클라우드 결과”



“인기상품의 트렌드 확인목적 활용가능”

“소비자 반응분석 프로그램 개발”



“판매자가 관심 상품에 대해 간편한 분석가능”

[판매상품의 소비자반응 피드백 / 신상품 출시 전 소비자반응 사전탐색]

예를 들어, 가구/인테리어 스토어팜 판매자라면, 가구/인테리어 구매평 빈도분석을 통하여 소비자가 의자나 조립식가구에 대해 수요가 높으며 색상과 가격특성 외에도 냄새특성(가구냄새)이 만족도를 결정하는 주요요인임을 고려하여 판매전략이나 신제품 런칭계획을 수립하는데 참고할 수 있다.

5. 한계점 및 향후 계획

“스토어팜 크롤링 수 제한”

트래픽과다 이유로 1회 URL접근 시,
크롤링 가능 구매평 1,000개 제한
데이터양(volume) 확보에 어려움



“네이버쇼핑 API권한 획득, 트래픽제한 회피”

“수동적 웹 크롤링”

일일이 URL을 입력하여 크롤링
데이터수집단계 작업시간 증가



“카테고리별 크롤링 자동화 구현”

“KOSAC 감성사전 부정어 처리”

부정어의 구매평 데이터 부족
KOSAC사전 및 분류 알고리즘 한계 등



“만족도가 ‘불만’인 구매평 수집확대,
KOSAC사전 및 분류 알고리즘 개선”

“판매량정보 등 활용정보 부족”

판매량과 클릭 수를 통한 인기순위만 제공
수치적인 분석결과 도출에 한계



“네이버쇼핑 API권한 개선요구
및 현업프로젝트 수행 시 판매량정보 요청”

Thank
You