빅데이터 분석과 응용

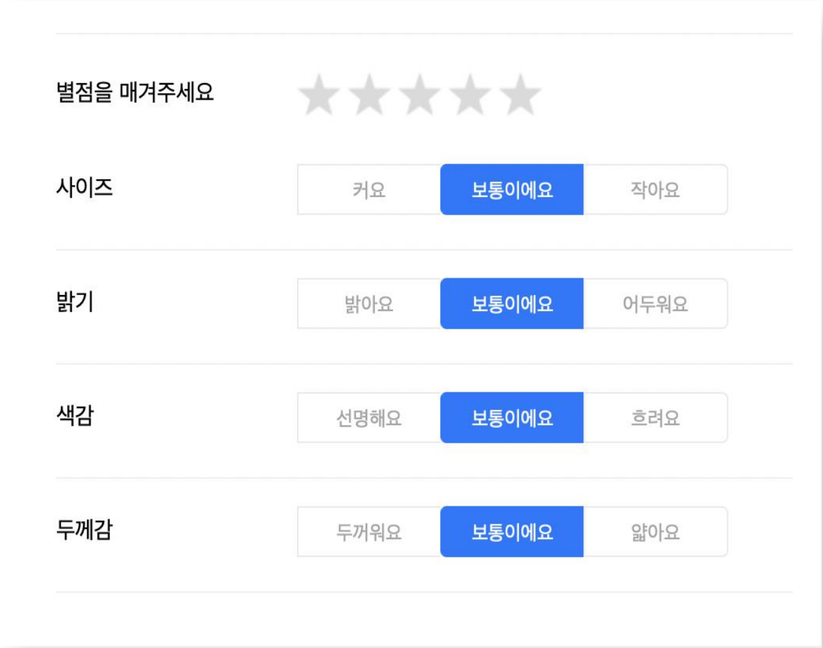
팀프로젝트 최종 발표

빅파이브(강예지, 권경민, 권혜준, 박수현, 변지민)

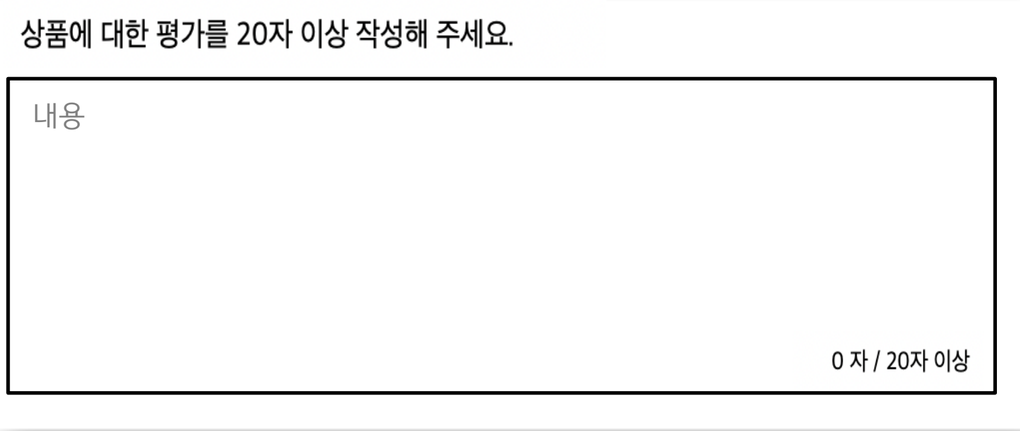
CONTENTS

문제정의

리뷰는 어떻게 작성되는가?



① 선택형

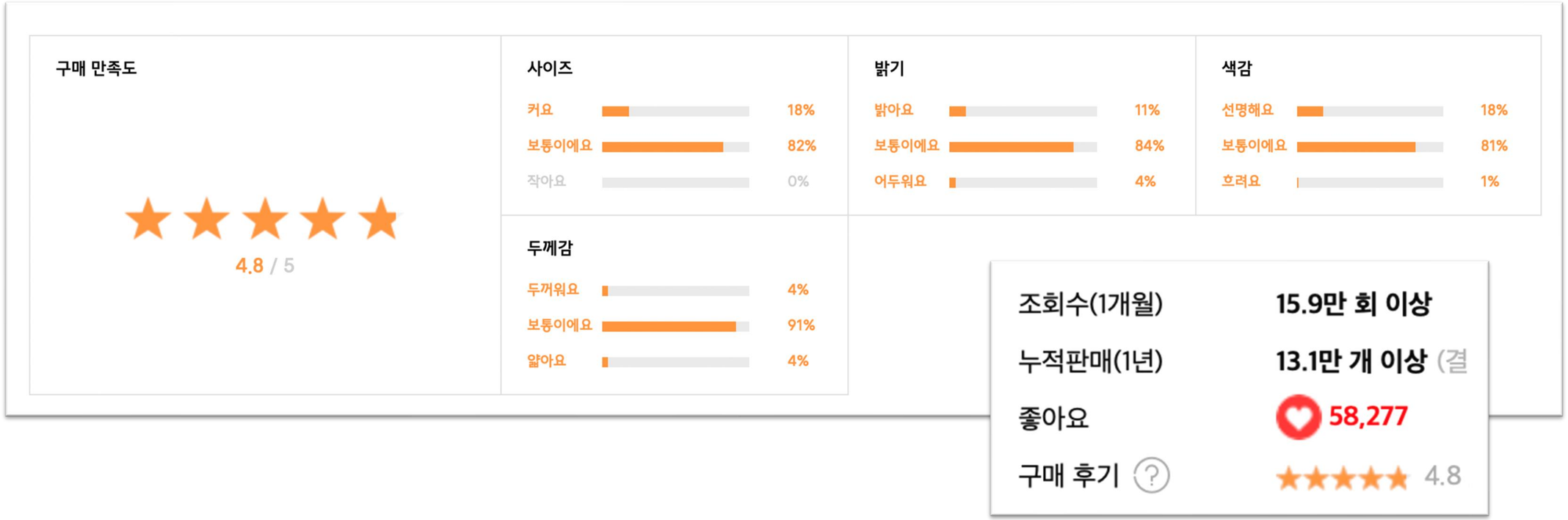


② 작성형

문제정의

선택형 리뷰작성의 데이터만 통계자료를 보여줌.

| 리뷰에 대한 통계 |



문제정의

선택형 리뷰작성의 데이터만 통계자료를 보여줌.

| 리뷰에 대한 통계

그렇다면 작성형(=리뷰) 자료는?

→ 소비자가 직접 봐야함.

0 1 주 제 선 정

MUSINSA 제품 리뷰 분석

작성형 후기의
분석이 필요함

MUSINSA

긍정리뷰 속
부정적인 내용찾기

데이터 수집

상품 사진 후기작성

- 작성하신 후기는 다른 회원들에게 공개됩니다. 댓글은 무신사에서 확인하지 않습니다.
- 상품 사진 후기 작성 시 1,000원의 적립금을 평일 기준 2일 전후로 지급합니다.
- 아래에 해당할 경우 적립금 지급이 보류되며, 이미 지급받으셨더라도 2차 검수를 통해 적립금을 회수할 수 있습니다. 또한 일부 후기는 조건에 따라 비노출 처리됩니다.
  - 포장이 제거되지 않았거나 상품의 전체 형태가 또렷하게 보이지 않는 후기
  - 상품을 직접 착용한 사진을 사용한 후기
  - 상품과 관련없거나 문자 및 기호의 단순 나열, 반복된 내용의 후기
  - 개인정보 및 광고, 비속어가 포함된 내용의 후기 (비노출 대상)
  - 상품 상세 페이지 등의 판매 이미지 사용, 관련없는 상품의 사진, 타인의 사진을 도용한 후기 (비노출 대상)
- 특히 후기 도용 시 적립금 2배 회수, 1년간 커뮤니티 이용 제한, 3개월간 후기 적립금 지급이 중단됩니다.
- 신체정보(성별, 키, 몸무게)는 수집 · 이용 동의 시 후기 서비스 제공 목적으로만 이용되며, 무신사 개인정보 처리방침에 따라 처리됩니다.

상품후기가 적절한 데이터임을 확인

0 2 데 이 터 수 집

크롤링

3개의 카테고리에서 3개의 후기순 정렬을 반영한 총 9개의 제품 후기

1. 후기순 정렬

2. 세 종류의 후기중 하나 선택

3. 별점과 텍스트

크롤링

자동화 이용

```
driver.find_element_by_xpath('//*[@id="estimate_photo"]').click()
```

별점과 리뷰 크롤링

```
element = driver.find_elements_by_css_selector('.review-list__rating__active')
width = element[j].get_attribute('style')
```

토픽 모델링

2. 토픽모델링: LDA

```
lda = LatentDirichletAllocation(n_components=20)  # 토픽 수는 20개로 설정
lda.fit(feat_vect)
```

```
LatentDirichletAllocation(n_components=20)
```

3. 토픽별 연관어 출력

```
def display_topics(model, feature_names, num_top_words):
    for topic_index, topic in enumerate(model.components_):
        print('Topic #', topic_index+1)

        # components_ array에서 가장 값이 큰 순으로 정렬했을 때, 그 값의 array index를 반환.
        topic_word_indexes = topic.argsort()[::-1]
        top_indexes=topic_word_indexes[:num_top_words]

        # top_indexes대상인 index별로 feature_names에 해당하는 word feature 추출 후 join으로 concat
        feature_concat = ' '.join([feature_names[i] for i in top_indexes])
        print(feature_concat)

# CountVectorizer객체내의 전체 word들의 명칭을 get_features_names( )를 통해 추출
feature_names = count_vectorizer.get_feature_names()

# Topic별 가장 연관도가 높은 word를 10개만 추출
display_topics(lda, feature_names, 10)
```

```
Topic # 1
여름 두께 정도 길이 편이 조금 일단 코튼 재질 여름 비치
Topic # 2
정말 마음 스탠다드 듭니 이즈 정사 마음 듭니 정사 이즈 역시 역시 스탠다드
Topic # 3
제품 색도 역시 의사 브랜드 다른 강추 그루 브라 무조건
Topic # 4
추천 색감 만족 드네 다운 렉스 사이즈 다운 항상 마음 드네 마음
Topic # 5
티셔츠 완전 여름 일리 기본 자체 어디 기본 티셔츠 가을 사람
```

분산된 토픽 키워드

토픽 모델링은 예측치로 값이 돌릴 때마다 달라져 정확한 결과를 도출하기 어렵고, 토픽이 분산되어 원하는 토픽을 도출하여 분류하기 어려움.

변덕꾸러기 토픽모델링 어떻게 다뤄야 하나

토픽 모델링 인공지능이 학습하는 방법에 '무작위'(랜덤) 요소가 들어있기 때문이다. 처음엔 모든 단어를 무작위 토픽에 배정하였으니 당연히 배정된 토픽은 이상할 것이다.

출처 : 한겨레

03 데이터 분석

워드클라우드

티셔츠

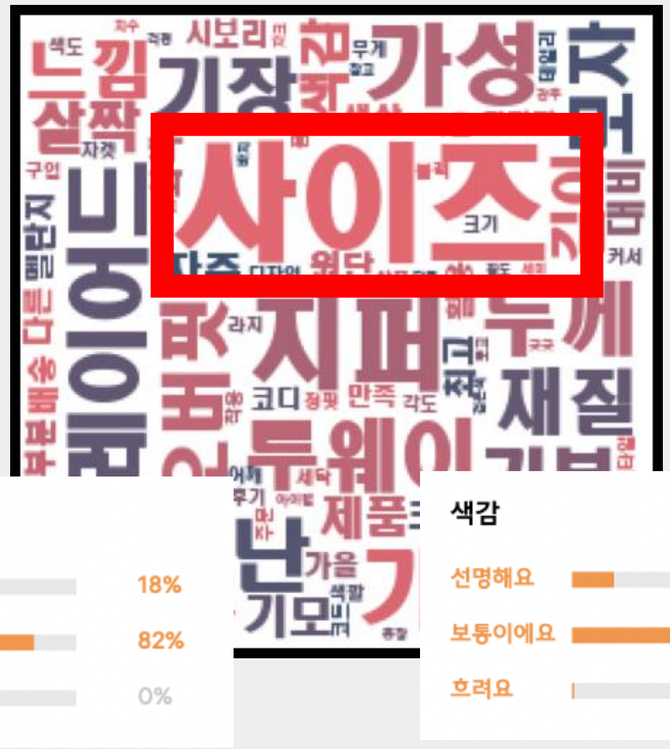
후드

청바지

워드클라우드

티셔츠

후드



청바지

리뷰에서 높은 빈도를 차지하는 키워드 -> 무신사에서 가이드를 제공한다는 사실을 알 수 있었음

03 데이터 분석

감성분석

1. 훈련된 모델인 pororo를 이용하여 크롤링을 통해 불러온 데이터 중 상품 후기들을 한국어 형태소로 분석.

```
[5]  import time
     from pororo import Pororo

[6]  sa=Pororo(task='sentiment',lang='ko') # 한국어 형태로 분석

     pos_prob, neg_prob=0.0, 0.0
     for comment in review1['상품후기']: #리뷰내용 칼럼에 있는 리뷰들 분석
       sa_result=sa(comment[0],show_probs=True)

[7]  sa = Pororo(task="sentiment", model="brainbert.base.ko.shopping", lang="ko")
     #감성분석

[8]  df=review1['상품후기'] #후기내용만 출력함
     df.head()

     0            디자인, 색상, 배송, 재질 모두 만족합니다. 추천합니다.
     1            디자인, 색상, 배송, 재질 모두 만족합니다. 추천합니다.
     2        배송 빠르고 재질도 좋아요! 사이즈는 음 살짝 큰느낌이 잇네요!
     3    색깔이 생각했던것보다 아주 조금 밝은거같지만Wn재질이랑 사이즈 다 좋네요
     4         아주시원한 재질은 아니지만 저렴하게 사서 만족하고 잘입을듯합니다
     Name: 상품후기, dtype: object
```

```
[10] x=[] # 결과를 담기 위해 리스트 형태로 사전에 저장함
     y=[] # 결과를 담기 위해 리스트 형태로 사전에 저장함
     for r in range(0,len(review1)): #처음부터 끝까지 긍정인 후기와 부정인 후기를 분류함
       result_dict=sa(df[r],show_probs=True)

       neg_prob=result_dict['negative'] # Negative인 경우 neg_prob에 포함
       pos_prob=result_dict['positive'] # Positive인 경우 pos_prob에 포함

       x.append(pos_prob) # pos_prob에 하나씩 추가
       x
       y.append(neg_prob) # neg_prob에 하나씩 추가
       y

     review1['긍정확률']=x
     review1['부정확률']=y

     review1['최종판단']=review1.apply(lambda i : 'Positive' if i['긍정확률']>i['부정확률'] else 'Negative',axis=1 )
      # 긍정확률 > 부정확률이면 Positive, 부정확률 > 긍정확률은 Negative 출력
     review1.head()
```

	상품후기	별점	긍정확률	부정확률	최종판단
0	디자인, 색상, 배송, 재질 모두 만족합니다. 추천합니다.	5	0.991129	0.008871	Positive
1	디자인, 색상, 배송, 재질 모두 만족합니다. 추천합니다.	5	0.991129	0.008871	Positive
2	배송 빠르고 재질도 좋아요! 사이즈는 음 살짝 큰느낌이 잇네요!	4	0.992625	0.007375	Positive
3	색깔이 생각했던것보다 아주 조금 밝은거같지만Wn재질이랑 사이즈 다 좋네요	5	0.986901	0.013099	Positive
4	아주시원한 재질은 아니지만 저렴하게 사서 만족하고 잘입을듯합니다	5	0.978576	0.021424	Positive

```
[ ]  review1.to_excel('감성분석_티1.xlsx',index=False)
```

2. 후기들을 감성분석하였고 분석 결과 긍정확률 > 부정확률이라면 positive로, 판단하고 반대의 경우 negative로 판단함.

03 데이터 분석

검색기능구현

1. 중복 리뷰 제거한 후, 알고 싶은 키워드 검색하고 긍정과 부정 선택하기

```
review_df = pd.read_excel('./감성분석_티1.xlsx') # 데이터 위치 주의
review_df = review_df.drop_duplicates(['상품후기'], ignore_index = True)
#중복 리뷰 제거(동일한 상품을 여러개 산 고객이 같은 내용으로 리뷰를 다는 경우
```

```
keyword = input("알고 싶은 키워드를 입력해주세요:")
```

```
알고 싶은 키워드를 입력해주세요:넥라인
```

```
review_df = review_df[review_df['상품후기'].str.contains(keyword)]
review_df
```

```
review_view = input("'긍정'과 '부정' 중에 선택해주세요.:")
```

```
'긍정'과 '부정' 중에 선택해주세요.:긍정
```

2. 긍정확률과 부정확률이 높은 순으로 정렬하여 원하는 후기 출력하기

```
Positive_review = review_df.loc[review_df['최종판단'] == 'Positive']
Negative_review = review_df.loc[review_df['최종판단'] == 'Negative']

if review_view == '긍정':
    Positive_review = Positive_review.sort_values(by = '긍정확률', ascending = False)
    #긍정확률 높은 순으로 긍정후기 출력하기
    print('< 긍정 리뷰 모아보기 >')
    for i in range(len(Positive_review)):
        print('----------------------------------------------------')
        print(i+1,':', Positive_review['상품후기'].iloc[i])
elif review_view == '부정':
    Negative_review = Negative_review.sort_values(by = '부정확률', ascending = False)
    #부정확률 높은 순으로 부정후기 출력하기
    print('< 부정 리뷰 모아보기 >')
    for i in range(len(Negative_review)):
        print('----------------------------------------------------')
        print(i+1,':', Negative_review['상품후기'].iloc[i])
else:
    print('잘못된 값입니다.')
```

03 데이터 분석

검색기능구현

3. 긍정 부정 리뷰 개수와 비율 구하기

```
Positive_num = len(Positive_review) #긍정리뷰 개수
Positive = Positive_num/len(review_df)*100
Positive = round(Positive, 2) #긍정리뷰 비율(소숫점 2자리 반올림)


Negative_num = len(Negative_review) #부정리뷰 개수
Negative = Negative_num/len(review_df)*100
Negative = round(Negative, 2) #부정리뷰 비율(소숫점 2자리 반올림)

print('긍정리뷰 개수:',Positive_num,'개')
print('긍정리뷰 비율:',Positive,'%')
print('----------------------------')
print('부정리뷰 개수:',Negative_num,'개')
print('부정리뷰 비율:',Negative,'%')
```

```
긍정리뷰 개수: 423 개
긍정리뷰 비율: 82.78 %
----------------------------
부정리뷰 개수: 88 개
부정리뷰 비율: 17.22 %
```

```
from pandas import DataFrame as df
df = df(data={'긍정':[Positive], '부정':[Negative]})
df = df.transpose()
df.columns = ['비율']
df #긍정 부정 비율을 데이터프레임에 저장하기
```

	비율
긍정	82.78
부정	17.22

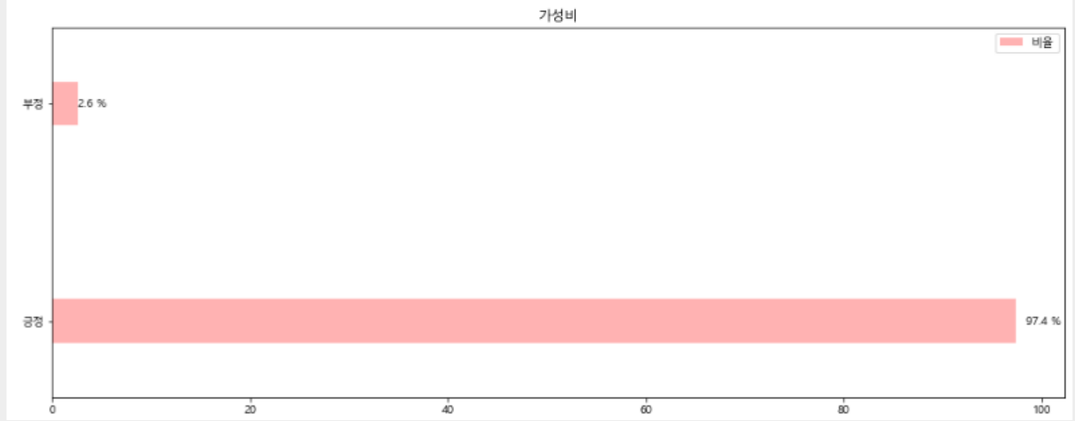
4. 긍정 부정 비율 시각화 하기

```
import matplotlib
from matplotlib import font_manager, rc
import platform

# 한글 깨짐 방지하기 위한 폰트 설정
if platform.system() == 'Windows':
    font_name = font_manager.FontProperties(fname="c:/Windows/Fonts/malgun.ttf").get_name()
    rc('font', family=font_name)
```

```
ax = df.plot(kind='barh', title=keyword, rot=0, figsize=(16,6), width=0.2, color=['red', 'blue'], alpha = 0.3)
#그래프 종류, 제목(keyword 변수) 등 지정

for p in ax.patches:
    x, y, width, height = p.get_bbox().bounds
    ax.text(width+1.01, y+height/2, "%.1f %%"%(width), va='center')
    #그래프 레이블 추가
```



시각화

1. 특정 키워드가 들어간 리뷰를 추출해 긍정 부정 리뷰 개수와 비율 구하기

```
review_df = pd.read_excel('./감성분석_티2.xlsx') # 데이터 위치 주의
review_df = review_df.drop_duplicates(['상품후기'], ignore_index = True)
#중복 리뷰 제거(동일한 상품을 여러개 산 고객이 같은 내용으로 리뷰를 다는 경우
review_df = review_df[review_df['상품후기'].str.contains("재질")] #재질이 포함된 리뷰 추출
```

```
Positive_num = len(review_df.loc[review_df['최종판단'] == 'Positive']) #긍정리뷰 개수
Positive = Positive_num/len(review_df)*100
Positive = round(Positive, 2) #긍정리뷰 비율(소숫점 2자리 반올림)

Negative_num = len(review_df.loc[review_df['최종판단'] == 'Negative']) #부정리뷰 개수
Negative = Negative_num/len(review_df)*100
Negative = round(Negative, 2) #부정리뷰 비율(소숫점 2자리 반올림)

print('긍정리뷰 개수:',Positive_num,'개')
print('긍정리뷰 비율:',Positive,'%')
print('--------------------------')
print('부정리뷰 개수:',Negative_num,'개')
print('부정리뷰 비율:',Negative,'%')
```

```
긍정리뷰 개수: 505 개
긍정리뷰 비율: 93.69 %
--------------------------
부정리뷰 개수: 34 개
부정리뷰 비율: 6.31 %
```

```
from pandas import DataFrame as df
df = df(data={'긍정':[Positive], '부정':[Negative]})
df = df.transpose()
df.columns = ['비율']
df #긍정 부정 비율을 데이터프레임에 저장하기
```

	비율
긍정	93.69
부정	6.31

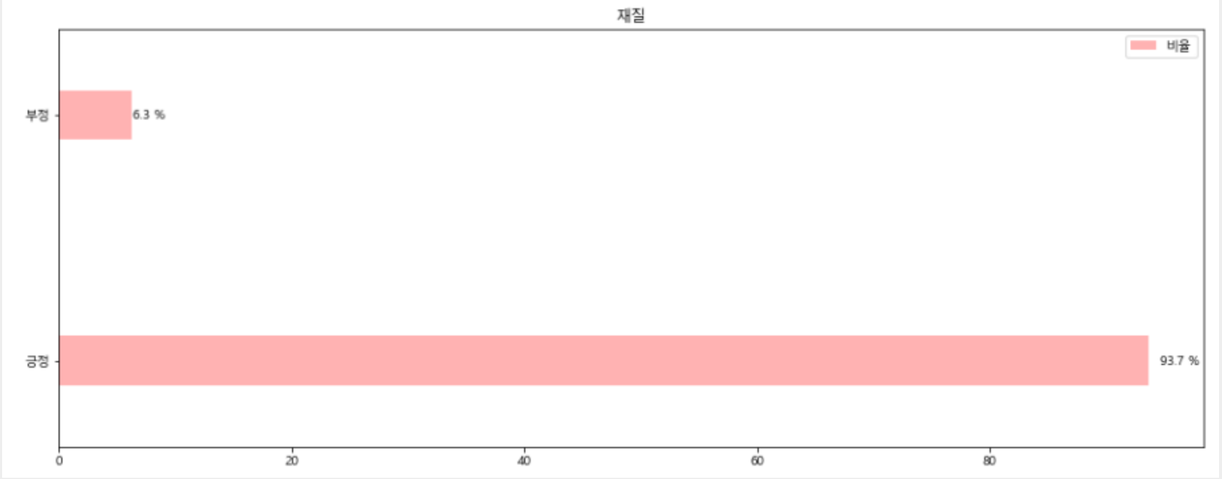
2. 긍정 부정 비율 시각화 하기

```
import matplotlib
from matplotlib import font_manager, rc
import platform

if platform.system() == 'Windows':
# 한글 깨짐 방지하기 위한 폰트 설정
    font_name = font_manager.FontProperties(fname="c:/Windows/Fonts/malgun.ttf").get_name()
    rc('font', family=font_name)
```

```
ax = df.plot(kind='barh', title="재질", rot=0, figsize=(16,6), width=0.2, color=['red', 'blue'], alpha = 0.3)
#그래프 종류, 제목 등 지정

for p in ax.patches:
    x, y, width, height = p.get_bbox().bounds
    ax.text(width+1.01, y+height/2, "%.1f %%"%(width), va='center')
    #그래프 레이블 추가
```

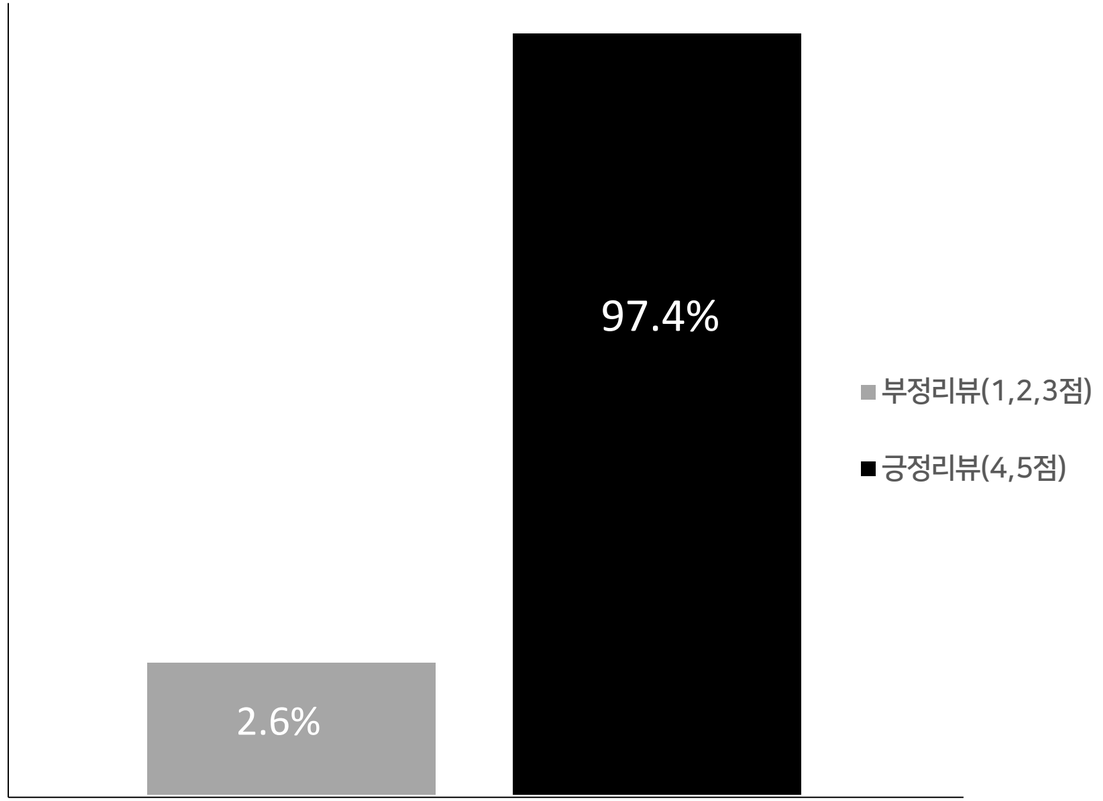


시각화

– 긍정적인 후기와 부정적인 후기 각각의 비율 –

제품	별점 1,2,3리뷰	별점 4,5리뷰
티셔츠1	3.1%	96.9%
티셔츠2	3.1%	96.9%
티셔츠3	1.8%	98.2%
후드1	2.6%	97.4%
후드2	2.4%	97.6%
후드3	1.2%	98.8%
청바지1	2.8%	97.2%
청바지2	2.7%	97.3%
청바지3	2.4%	97.6%

〈퍼센트 평균〉

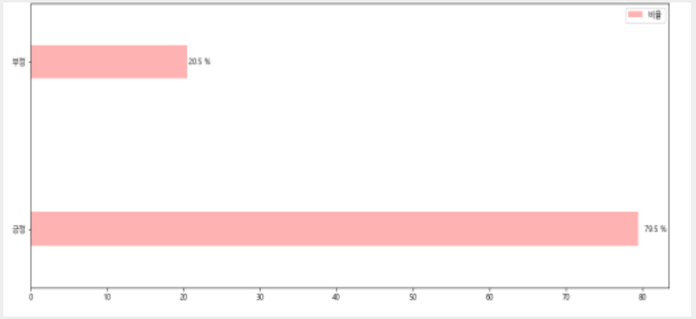


긍정리뷰가 압도적으로 많다.

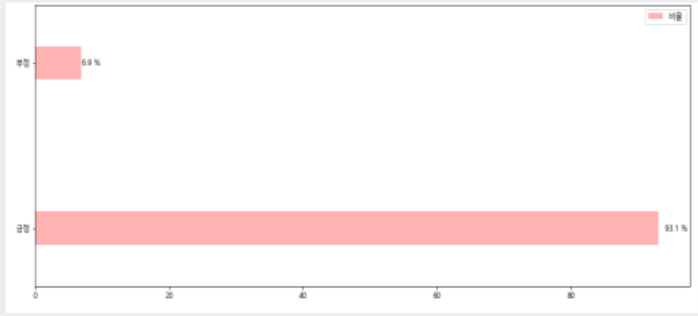
시각화 자료 – 티셔츠

티셔츠 1번_무신사 스탠다드

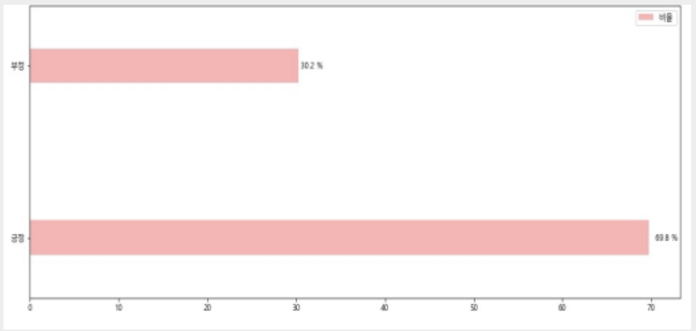
구김

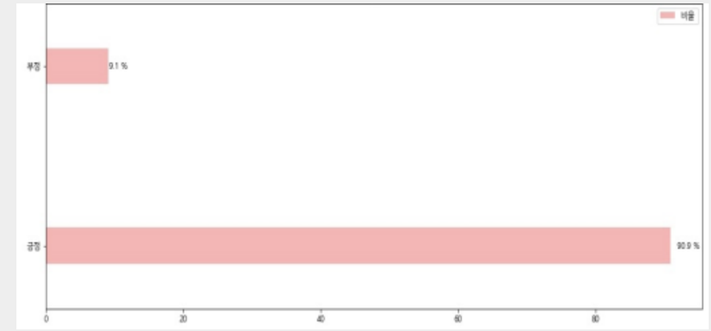


비침

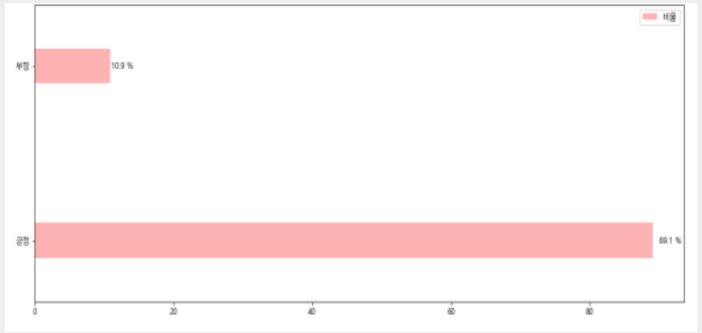


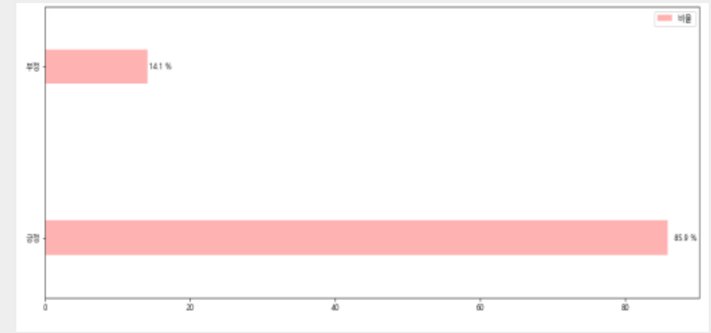
티셔츠 2번_그루브라임





티셔츠 3번_어커버

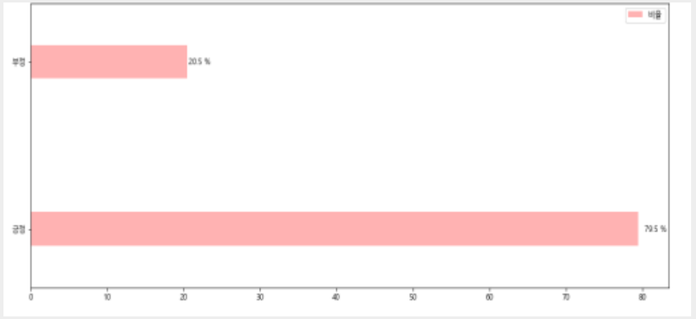




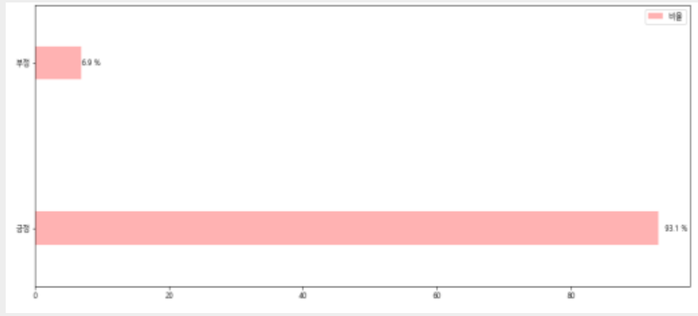
시각화 자료 – 티셔츠

티셔츠 1번_무신사 스탠다드

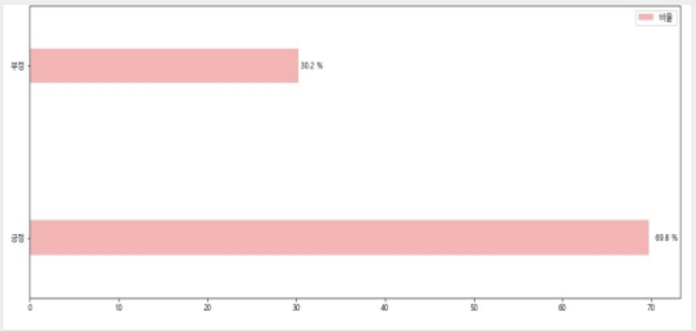
구김

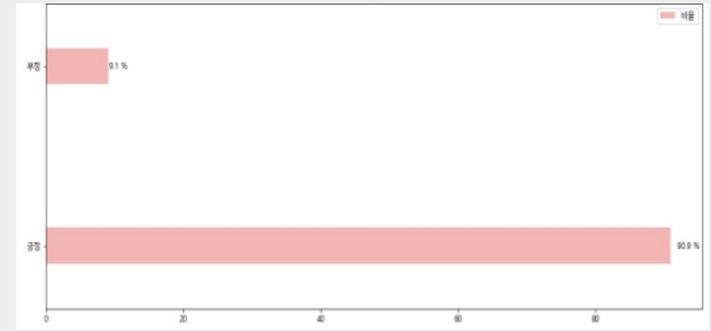


비침

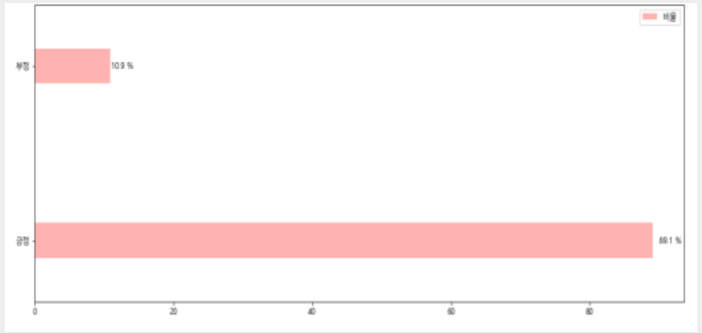


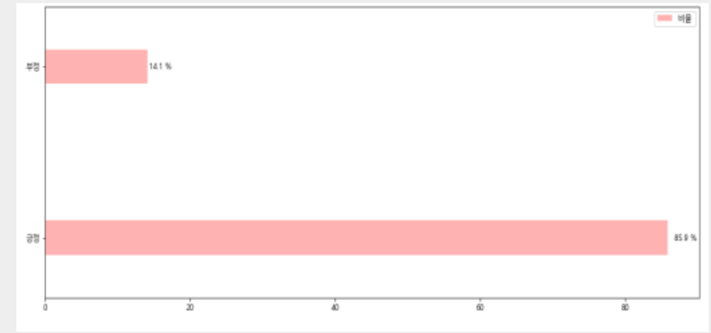
티셔츠 2번_그루브라임





티셔츠 3번_어커버





시각화 자료 – 티셔츠

구김이 적은 티셔츠
3번

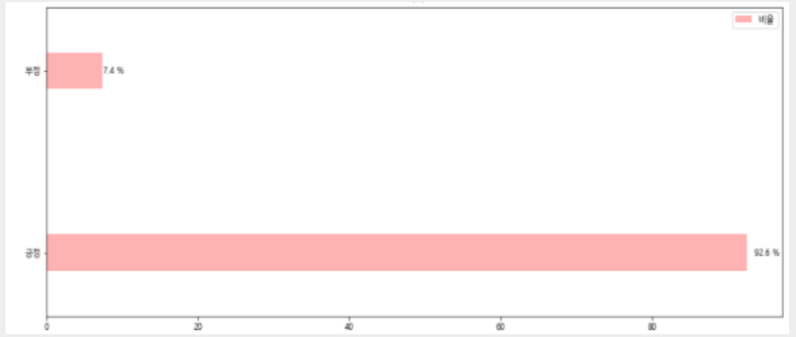
비침이 적은 티셔츠
1번

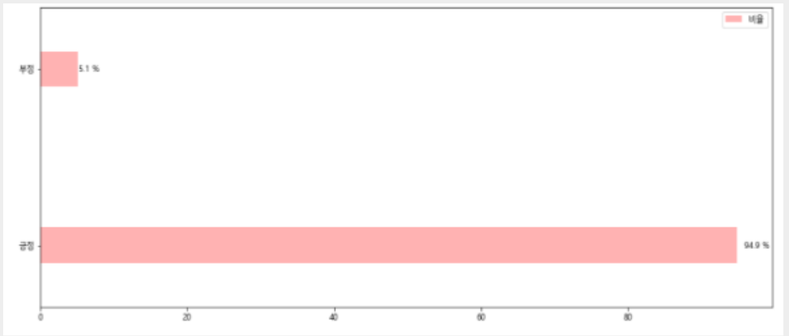
시각화 자료 – 후드

지퍼

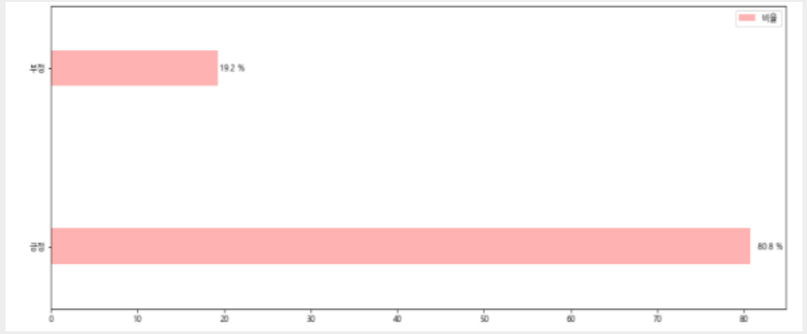
핏감

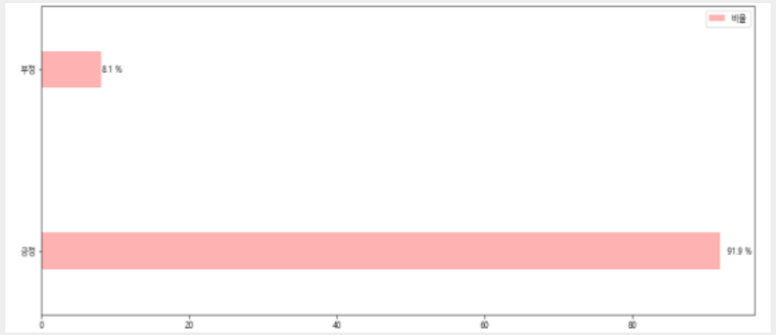
후드 1번_멜란지마스터



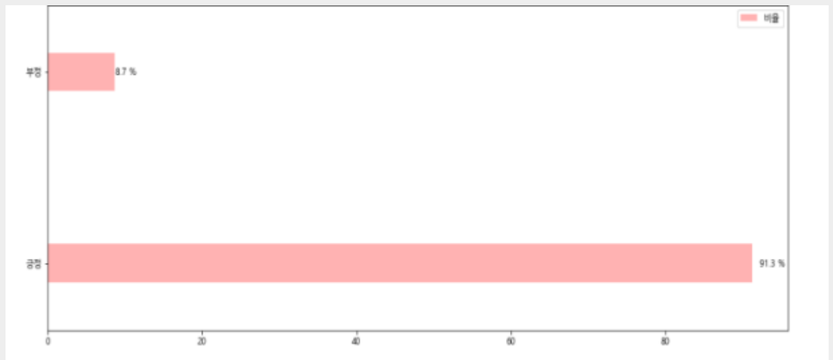


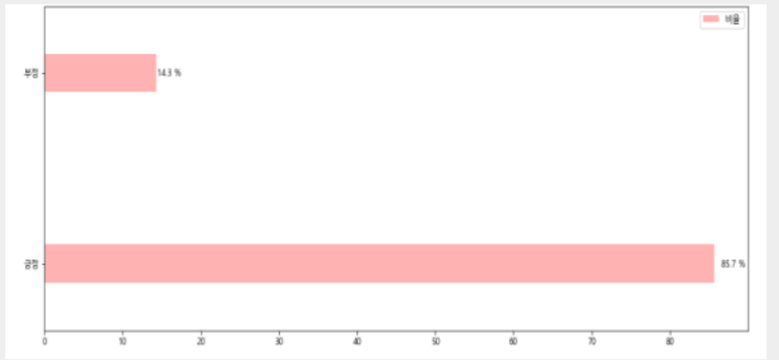
후드 2번_토피





후드 3번_무신사 스탠다드





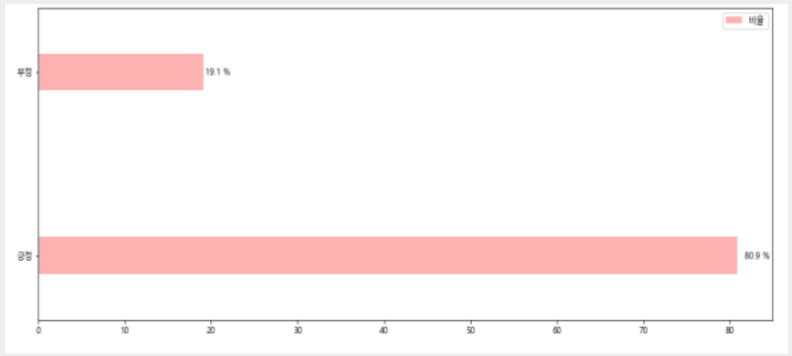
시각화 자료 – 후드

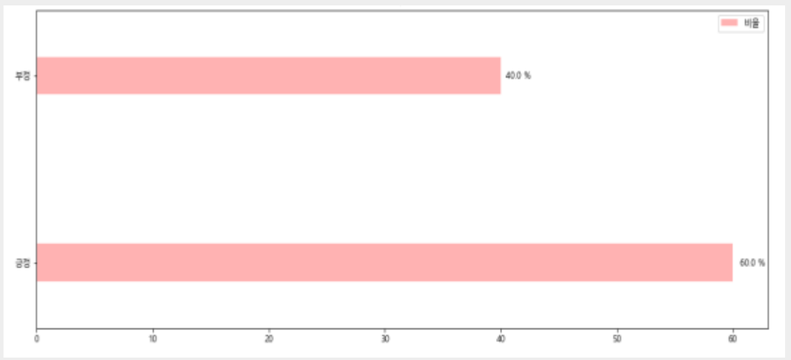
지퍼 퀄리티가 좋은 후드
1번

핏감이 좋은 후드
1번

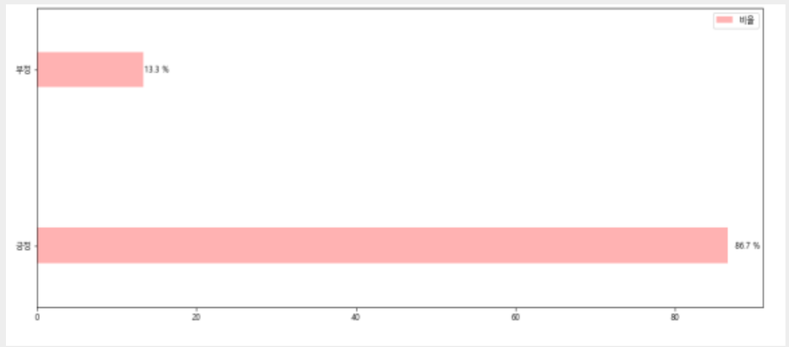
시각화 자료 – 청바지

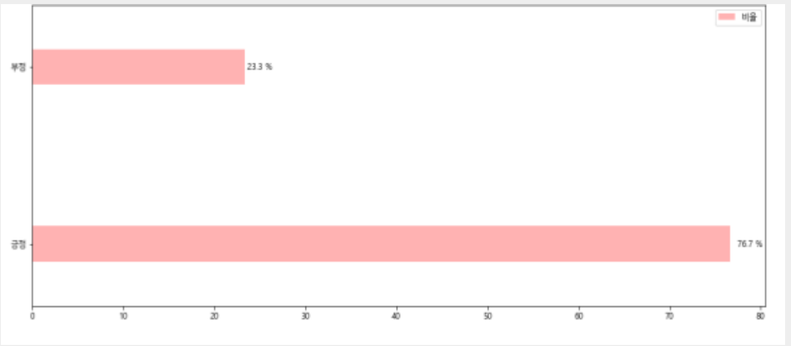
청바지 1번_토피



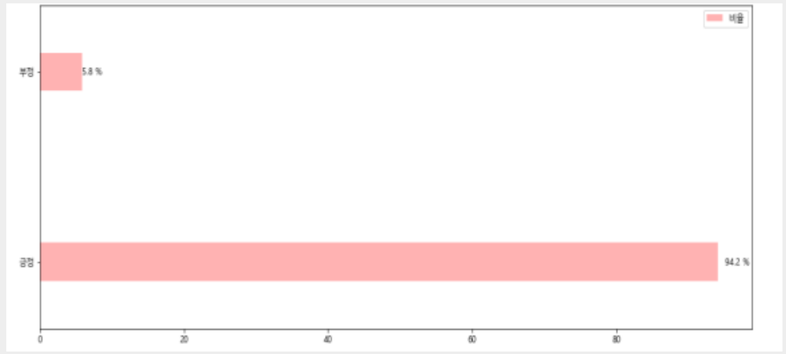


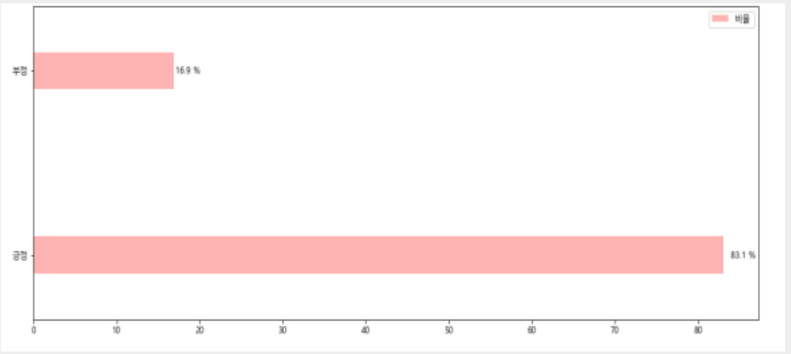
청바지 2번_페이탈리즘





청바지 3번_브랜디드





시각화 자료 – 청바지

워싱이 예쁜 청바지
3번

이염이 적은 티셔츠
3번

05 결론

MUSINSA

작성형 후기의
분석

워드 클라우드를 기반으로 하여
보편적인(빈도수 높은) 키워드를
소비자에게 제공 가능

긍정리뷰 속
부정적인 내용찾기

전체 감성 분석 엑셀 파일에서
keyword 변수를 지정하여
개인이 원하는 키워드에 대한
리뷰와 시각화 검색 가능

확장시킨 방향성

05 결론

1. 작성형 리뷰를 분석해 온라인 쇼핑 만족도를 향상에 기여함.

2. 높은 별점을 준 리뷰에도 부정적인 내용이 들어감.

재질 구김 비침 워싱 이염 지퍼 후드

0 8 참 고 문 헌

1. 김경미, 전병호, 강병구(고려대학교 대학원 디지털경영학과), 온라인쇼핑몰의 피드백 시스템 설계에 관한 연구, 한국경영정보학회2007년도 International Conference(2007), pp 372-377

2. 김근형(제주대학교), 오성열(제주산업정보대학), 온라인 고객리뷰 분석을 통한 시장세분화에 텍스트마이닝 기술을 적용하기 위한 방법론, 한국콘텐츠학회논문지 제9권 제8호(2009.08), pp272 – 284

3. 오석원, 진서훈(한국자료분석학회), 텍스트 마이닝을 이용한 쇼핑몰 구매후기 분석, Journal of The Korean Data Analysis SocietyJournal of The Korean Data Analysis Society 제14권 제1호(2012), pp 125 – 137

4. 이광호, 김남규(국민대학교), 딥러닝을 활용한 쇼핑몰 고객군별 리뷰 감성 키워드 비교 분석, 2019년 한국지능정보시스템학회 추계학술대회 초록집(2019.10), pp 11-12), pp 541-555

0 8 참 고 문 헌

5. 엄세웅, 한동훈, 임태민, 한용구(경희대학교), 의류 온라인 쇼핑몰 상품 후기 객관화를 위한 온라인 쇼핑몰 댓글 및 리뷰 분석, 한국정보과학회 2019 한국소프트웨어종합학술대회 논문집(2019.12), pp 1492-1494

6. 임명진, 김판구, 신주현(조선대학교), 리뷰의 의미적 토픽 분류를 적용한 감성 분석 모델, (사)한국스마트미디어학회(2020), vol.9, no.2, pp. 69-77

7. 한기향(건국대학교), 반팔 티셔츠의 온라인 리뷰 분석에 관한 융합적 분석 연구, 한국과학예술융합학회 Vol.39 No.4(2021.09), pp 541-555

Thank you for your attention!