

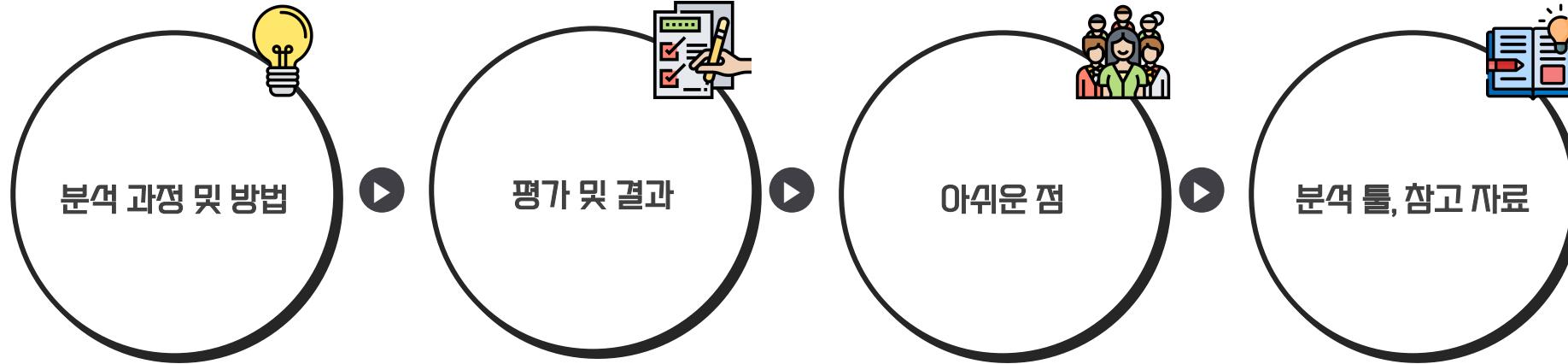
# 감성 분석을 통한 제품 소비자 리뷰 분석



빅데이터 분석 - 팀 프로젝트 결과 발표 PPT

소프트웨어학부 20170734 김유나

# CONTENTS



# 분석 과정 및 방법

STEP 1.

-크롤링으로 실제 리뷰 데이터 수집

## 네이버 쇼핑 HTML Tag 분석

```
<div class="reviewItems etc_area_2P8i3">
  <span class="reviewItems_average_16ya-a">
    <span class="reviewItems_star_2EEY8">···</span>
    "5"
  </span>
  <span class="reviewItems_etc_1YqVF">해라</span>
  <span class="reviewItems_etc_1YqVF">···</span>
  <span class="reviewItems_etc_1YqVF">···</span>
  <span class="reviewItems_etc_1YqVF">···</span>
</div>
<div class="reviewItems_review_1eF8A">
  <div class="reviewItems_review_text_2Bwpa">
    <em class="reviewItems_title_39ZBH">
      "捏 색상21호 노란끼있는 피부라서 21N1 구매했습니다! 21호피부에 딱 맞는 색상이에요! 적당히 희
      사한 느낌에 목이랑 너무 동떨어지지않는 자연스러운 느낌이에요. 기존 블랙쿠션 17"
    </em>
    <p class="reviewItems_text_XIsTc">
      <em>捏 색상</em>
      <br>
      "21호 노란끼있는 피부라서 21N1 구매했습니다!"
    <br>
    "21호피부에 딱 맞는 색상이에요! 적당히 화사한 느낌에 목이랑 너무 동떨어지지않는 자연스러운
    느낌이에요. 기존 블랙쿠션 17, 21, 23호 모두 사용해봤는데 21호와 동일한 컬러인것같아요! "
    <br>
    <em>밀착력이 너무 좋은제품이라 지속력이 정말 좋았어</em>
    "요! 기존 블랙쿠션도 지속력이 정말 좋은편이었는데 이번 제품은 밀착력이 훨씬 더 좋아진것같아
    요! 너무 두껍지않게 알게 발리면서 피부에 알게 밀착돼서 묻어남도 정말 적고 들뜸도 없어요! 시
    간이 지나도 끼임이나 들뜸도 없고 유분이 조금 올라오는것 제외하고는 무너짐도 거의 느껴지지않던
    제품이에요! 다른날도 있고 너무 만족스러웠습니다! 요즘 마스크를 매일 사용하는데 묻어남도 적고 미
    스크린에서 무너짐도 적어서 너무 좋아있어요!"
  <br>
```

## Selenium과 BeautifulSoup로 크롤러 구현

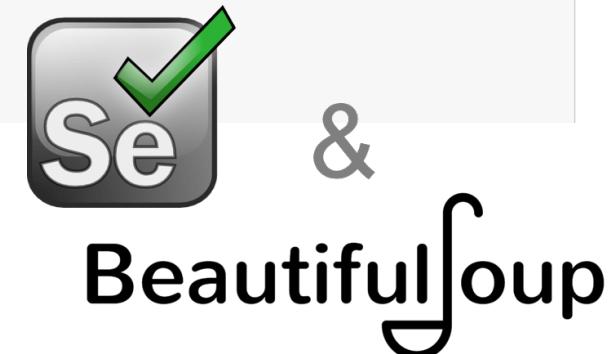
```
def get_reviews(self, num_of_total_page):
    driver = self.__driver
    product = self.__product
    driver.get(self.__url)
    driver.set_window_size(1920, 1080)

    print(f"{product} 크롤링 시작 -----")
    print(f"전체 페이지 : {num_of_total_page}")
    # 전체 페이지 개수만큼 크롤링
    for page in range(num_of_total_page):
        self.change_page(page+1)

        time.sleep(0.5)
        now_page_before_parse = driver.find_element_by_css_selector(
            '[class="reviewItems_list_review_lsgcj"] + .pagination_pagination_2M9a4 [class="pagination_now_gZWGF"
            num_of_now_page = int(
                now_page_before_parse.replace("현재 페이지", "")).strip()

        print(f"현재 크롤링 중인 페이지 : {num_of_now_page}")
        self.extract_review()

    self.__driver.quit()
    print("End Scraping")
    return self.__reviews
```





# 분석 과정 및 방법

STEP 1.

- 크롤링으로 실제 리뷰 데이터 수집
- 네이버 쇼핑 이십만개 데이터 가져와 전처리

## 감성 분석용 말뭉치

이 폴더에는 감성 분석에 사용할 수 있도록 긍정(긍정/부정)이 라벨링된 텍스트 데이터가 들어 있습니다. naver\_shopping과 steam 데이터는 간단한 감성 분류 실험을 위해서 수집된 것으로 실행됩니다.

### naver\_shopping.txt

- 언어: 한국어
- 출처: 네이버 쇼핑 (<https://shopping.naver.com/>)
- 수집 기간: 2020.06~2020.07
- 데이터 건수: 20만 건

네이버 쇼핑에서 제품별 후기를 별점과 함께 수집한 것입니다. 데이터는 탭으로 분리되어 있으므로 텍스트가 위치합니다. 긍/부정으로 분류하기 애매한 3점에 해당하는 텍스트들은 제외되어 비율이 1:1에 가깝도록 샘플링하였습니다.

https://github.com/bab2min/corpus/tree/master/sentiment  
네이버 쇼핑 리뷰 데이터 가져와 중복 데이터, 전처리, 결측값 제거 후 csv로 저장

```
In [5]: # https://github.com/bab2min/corpus/tree/master/sentiment 네이버 쇼핑 리뷰 데이터 가져오기
urllib.request.urlretrieve("https://raw.githubusercontent.com/bab2min/corpus/master/sentiment/naver_shopping.txt", file)
total_data = pd.read_table('ratings_total.txt', names = ['grade', 'reviews'])
```

```
In [6]: # 긍정, 부정 라벨 추가
total_data['label'] = np.select([total_data.grade > 3], [1], default = 0)
total_data[:5]
```

Out[6]:

	grade	reviews	label
0	5	배공빠르고 굿	1
1	2	택배가 엉망이네용 저희집 밑에총에 말도없이 놔두고가고	0
2	5	아주좋아요 바지 정말 좋아서2개 더 구매했어요 이가격에 대박입니다. 바느질이 조금 ...	1
3	2	선물용으로 빨리 받아서 전달했어야 하는 상품이었는데 머그컵만 와서 당황했습니다. 전...	0
4	5	민트색상 예뻐요. 옆 손잡이는 거는 용도로도 사용되네요 ㅎㅎ	1

```
In [7]: # 중복 데이터 제거
total_data.drop_duplicates(subset = ['reviews'], inplace = True)
```

```
In [8]: # 한글과 공백제외하고 모두 제거
```

```
total_data['reviews'] = total_data['reviews'].str.replace("[^ㄱ-ㅎㅏ-ㅣ가-힣 ]", "")
total_data['reviews'].replace('', np.nan, inplace=True) # 공백 null 처리
total_data = total_data.dropna(how='any') # null 값 제거
```

# 분석 과정 및 방법

STEP 1.

- 크롤링으로 실제 리뷰 데이터 수집
- 크롤링을 통해 실제 리뷰 데이터 수집 및 전처리
  - 실제 리뷰 데이터 전처리

## 실제 리뷰 데이터 전처리 함수 구현

```
# 실제 쇼핑몰 리뷰 데이터 전처리
hera_data = pd.read_csv('헤라블랙쿠션_review.csv')
doctorG_data = pd.read_csv('닥터지마일드선크림_review.csv')
buds_data = pd.read_csv('삼성갤럭시버즈_review.csv')

def clean_data(df):
    df['제목'] = df['제목'].str.replace("[^ㄱ-ㅎㅏ-ㅣ가-힣]", "")
    df['제목'].replace('', np.nan, inplace=True) # 공백 null 처리
    df['내용'] = df['내용'].str.replace("[^ㄱ-ㅎㅏ-ㅣ가-힣]", "")
    df['내용'].replace('', np.nan, inplace=True) # 공백 null 처리
    df = df.dropna(how='any') # null 값 제거
    return df
```

## 실제 리뷰 데이터 전처리 비교

hera\_data[:5]

평점	날짜	제목	내용
0	5 21.04.14.	전 복합성피부 입니다. 코 부분은 엄청 피지 뿐뿐인데 눈가 이마 볼은 건조해요.저는...	전 복합성피부 입니다. 코 부분은 엄청 피지 뿐뿐인데 눈가 이마 볼은 건조해요.\n...
1	5 21.03.31.	☞ 색상21호 노란끼있는 피부라서 21N1 구매했습니다!21호피부에 딱 맞는 색상이에...	☞ 색상\n21호 노란끼있는 피부라서 21N1 구매했습니다!\n21호피부에 딱 맞는 ...
2	3 21.05.07.	배송이느려도 넘느려서 기다리기 지치고 내려놓무렵에야 왔네요ㅠㅠ4.13일날 주문해서 ...	배송이느려도 넘느려서 기다리기 지치고 내려놓무렵에야 왔네요ㅠㅠ4.13일날 주문해서 ...
3	5 21.04.01.	리뉴얼 전 블랙쿠션도 아주 잘쓰고있었습니다.엄청 건조한 겨울만 아니면 거의 블랙쿠션...	리뉴얼 전 블랙쿠션도 아주 잘쓰고있었습니다.\n엄청 건조한 겨울만 아니면 거의 블랙...
4	5 21.05.23.	헤라 쿠션팩트는 정말 갖고 싶었는데 3번째 라이브방송만에 드디어 구매했습니다! 다른...	헤라 쿠션팩트는 정말 갖고 싶었는데 3번째 라이브방송만에 드디어 구매했습니다! 다른...

```
hera_data = clean_data(hera_data)
hera_data[:5]
```



평점	날짜	제목	내용
0	5 21.04.14.	전 복합성피부 입니다 코 부분은 엄청 피지 뿐뿐인데 눈가 이마 볼은 건조해요저는...	전 복합성피부 입니다 코 부분은 엄청 피지 뿐뿐인데 눈가 이마 볼은 건조해요저는...
1	5 21.03.31.	색상호 노란끼있는 피부라서 구매했습니다호피부에 딱 맞는 색상이에요 적당히 화사한 ...	색상호 노란끼있는 피부라서 구매했습니다호피부에 딱 맞는 색상이에요 적당히 화사한 ...
2	3 21.05.07.	배송이느려도 넘느려서 기다리기 지치고 내려놓무렵에야 왔네요ㅠㅠ일날 주문해서 월 일 에...	배송이느려도 넘느려서 기다리기 지치고 내려놓무렵에야 왔네요ㅠㅠ일날 주문해서 월 일 에...
3	5 21.04.01.	리뉴얼 전 블랙쿠션도 아주 잘쓰고있었습니다엄청 건조한 겨울만 아니면 거의 블랙쿠션 만...	리뉴얼 전 블랙쿠션도 아주 잘쓰고있었습니다엄청 건조한 겨울만 아니면 거의 블랙쿠션 만...
4	5 21.05.23.	헤라 쿠션팩트는 정말 갖고 싶었는데 번째 라이브방송만에 드디어 구매했습니다 다른데 에...	헤라 쿠션팩트는 정말 갖고 싶었는데 번째 라이브방송만에 드디어 구매했습니다 다른데 에...

01

# 분석 과정 및 방법

STEP 2.

- koBERT 자연어 처리 및 BertClassificationModel 학습 후 평가

전처리한 데이터를 깃허브에 저장 후, 불러옴

```
# 미리 github에 올려놓은 데이터 파일 다운로드 (review_dataset, 실제 쇼핑몰 리뷰 크롤링 dataset)
!git clone https://github.com/yunakim2/BigDataAnalasisTermProject.git
os.listdir('BigDataAnalasisTermProject/processingdata')
```

```
Cloning into 'BigDataAnalasisTermProject'...
remote: Enumerating objects: 30, done.
remote: Counting objects: 100% (30/30), done.
remote: Compressing objects: 100% (22/22), done.
remote: Total 30 (delta 9), reused 25 (delta 7), pack-reused 0
Unpacking objects: 100% (30/30), done.
['naver_shopping.csv',
'doctorG_review.csv',
'buds_review.csv',
'hera_review.csv']
```



리뷰 데이터셋 train/test으로 나눔  
 (전체 데이터 개수가 199908 중,  
 랜덤으로 29986개 추출 후 train : test = 7: 3 으로 나눔)

```
data = pd.read_csv('BigDataAnalasisTermProject/processingdata/naver_shopping.csv')
print('전체 데이터 개수 : ', len(data))
total_data = data.sample(frac=0.15)
print('랜덤 추출 데이터 개수: ', len(total_data))
```

```
size = int(len(total_data) // 3)
print('train 데이터 개수 : ', size)
print('test 데이터 개수 : ', len(total_data) - size)
test = total_data[:size]
train = total_data[size:]
```

```
전체 데이터 개수 : 199908
랜덤 추출 데이터 개수: 29986
train 데이터 개수 : 9995
test 데이터 개수 : 19991
```

01

## 분석 과정 및 방법

## STEP 2.

- koBERT 자연어 처리 및 BertClassificationModel 학습 후 평가

## 리뷰 데이터 BertClassificationModel input 으로 변환 (koBERT로 토크나이즈된 데이터를 토큰, 세그먼트, 마스크 *input* 으로 변환)

```
▶ def convert_data(data_df):
    global tokenizer

    SEQ_LEN = 128 #SEQ_LEN : 버트에 들어갈 인풋의 길이

    tokens, masks, segments, targets = [], [], [], []

    for i in tqdm(range(len(data_df))):
        # token : 문장을 토큰화함
        token = tokenizer.encode(data_df.iloc[i]['reviews'], truncation=True, padding='max_length')

        # 마스크는 토큰화한 문장에서 패딩이 아닌 부분은 1, 패딩인 부분은 0으로 통일
        num_zeros = token.count(0)
        mask = [1]*(SEQ_LEN-num_zeros) + [0]*num_zeros

        # 문장의 전후관계를 구분해주는 세그먼트는 문장이 1개밖에 없으므로 모두 0
        segment = [0]*SEQ_LEN

        # 버트 인풋으로 들어가는 token, mask, segment를 tokens, segments에 각각 저장
        tokens.append(token)
        masks.append(mask)
        segments.append(segment)

        # 정답(긍정 : 1 부정 0)을 targets 변수에 저장해 줌
        targets.append(data_df.iloc[i][LABEL_COLUMN])

    # tokens, masks, segments, 정답 변수 targets를 numpy array로 지정
    tokens = np.array(tokens)
    masks = np.array(masks)
    segments = np.array(segments)
    targets = np.array(targets)

    return [tokens, masks, segments], targets

# 위에 정의한 convert_data 함수를 불러오는 함수를 정의
def load_data(pandas_dataframe):
    data_df = pandas_dataframe
    data_df[DATA_COLUMN] = data_df[DATA_COLUMN].astype(str)
    data_df[LABEL_COLUMN] = data_df[LABEL_COLUMN].astype(int)
    data_x, data_y = convert_data(data_df)
    return data_x, data_y
```

## 1. tokenizer 예시

- `tokenizer.tokenize` => 문장을 토큰화
  - `tokenizer.encode` => 문장을 버트 모델의 인풋 토큰값으로 변경

```
[ ] print(tokenizer.tokenize('너무 좋은 제품이예요. 또 구매하고 싶어요.'))  
[ '_너무', '_좋은', '_제품', '이', '예', '요', '.', '_또', '_구매', '하고', '_싶어', '요', '.']  
  
[ ] print(tokenizer.encode('너무 좋은 제품이예요. 또 구매하고 싶어요.'))  
[ 2, 1458, 4209, 4158, 7096, 6957, 6999, 54, 1861, 1119, 7788, 3073, 6999, 54, 3]
```

## 1. 토큰 인풋 변화

문장 토크나이징 후 `tokenizer.encode()` 결과 값으로 문장이 유효한 값이면 1로, 유효하지 않으면 0으로 채워 문장 길이가 다르지만 베트의 인풋길이를 일정하게 고정하기 위해 베트에서 지정한 문자 길이를 초과하면 패딩값 0으로 채운다.

## 2. 세그멘트 인풋 변환

세그멘트는 bert 모형에 들어가 bert 모형에 맞게 고차원으로 임베딩 되는 워리언

버트 모델에서 문장이 앞 문장인지 뒤 문장인지 표현해주는 것이고 지금 사용하고 있는 **dataset**은 한문장 이므로 '0'으로 통일한다.

최대 길이를 128로 고정했으므로 세그먼트도  $[0..128]$  가 되어야 한다.

```
[1] print([0]*128)
```

### 3. 마스크 인풋

токен 인풋에서 패딩이 아닌 부분은 1, 패딩인 부분은 0으로 두게 되는 것

```
[ ] valid_num = len(tokenizer.encode('너무 좋은 제품이예요. 또 구매하고 싶어요'))  
print(valid_num * [1] + (128-valid_num) * [0])
```



STEP 2.

- koBERT 자연어 처리 및 BertClassificationModel 학습 후 평가

BertClassificationModel train 데이터 셋으로 학습 후 test 데이터 셋으로 모델 평가

```
model = TFBertModel.from_pretrained("monologg/kobert", from_pt=True)
# 토큰 인풋, 마스크 인풋, 세그먼트 인풋 정의
token_inputs = tf.keras.layers.Input((SEQ_LEN,), dtype=tf.int32, name='input_word_ids')
mask_inputs = tf.keras.layers.Input((SEQ_LEN,), dtype=tf.int32, name='input_masks')
segment_inputs = tf.keras.layers.Input((SEQ_LEN,), dtype=tf.int32, name='input_segment')
# 인풋이 [토큰, 마스크, 세그먼트]인 모델 정의
bert_outputs = model([token_inputs, mask_inputs, segment_inputs])
```

모델 학습 과정

```
sentiment_model.fit(train_x, train_y, epochs=2, shuffle=True, batch_size=64, validation_data=(test_x, test_y))
```

```
Epoch 1/2
WARNING:tensorflow:The parameters `output_attentions`, `output_hidden_states` and `use_cache` cannot be updated when calling a model.
WARNING:tensorflow:The parameter `return_dict` cannot be set in graph mode and will always be set to `True`.
WARNING:tensorflow:The parameters `output_attentions`, `output_hidden_states` and `use_cache` cannot be updated when calling a model.
WARNING:tensorflow:The parameter `return_dict` cannot be set in graph mode and will always be set to `True`.
313/313 [=====] - ETA: 0s - loss: 0.6157 - accuracy: 0.6397WARNING:tensorflow:The parameters `output_attentiv
WARNING:tensorflow:The parameter `return_dict` cannot be set in graph mode and will always be set to `True`.
313/313 [=====] - 390s 1s/step - loss: 0.6157 - accuracy: 0.6397 - val_loss: 0.3573 - val_accuracy: 0.8630
Epoch 2/2
313/313 [=====] - 338s 1s/step - loss: 0.3177 - accuracy: 0.8799 - val_loss: 0.3007 - val_accuracy: 0.8900
<tensorflow.python.keras.callbacks.History at 0x7fe31e749a10>
```

Test 데이터 셋으로 평가 결과

```
from sklearn.metrics import classification_report
y_true = test['label']
# F1 Score 확인
print(classification_report(y_true, np.round(preds,0)))
```

	precision	recall	f1-score	support
0	0.94	0.83	0.88	4991
1	0.85	0.95	0.90	5004
<hr/>				
accuracy			0.89	9995
macro avg	0.90	0.89	0.89	9995
weighted avg	0.90	0.89	0.89	9995

# 분석 과정 및 방법

STEP 3.

- 실제 리뷰 데이터 감성 분류 (긍정 리뷰, 부정 리뷰)

```
def sentence_convert_data(data):
    global tokenizer
    tokens, masks, segments = [], [], []
    token = tokenizer.encode(data, max_length=SEQ_LEN, truncation=True, padding='max_length')

    num_zeros = token.count(0)
    mask = [1]*(SEQ_LEN-num_zeros) + [0]*num_zeros
    segment = [0]*SEQ_LEN

    tokens.append(token)
    segments.append(segment)
    masks.append(mask)

    tokens = np.array(tokens)
    masks = np.array(masks)
    segments = np.array(segments)
    return [tokens, masks, segments]

def review_evaluation_predict(sentence):
    data_x = sentence_convert_data(sentence)
    predict = sentiment_model.predict(data_x)
    predict_value = np.ravel(predict)
    predict_answer = np.round(predict_value,0).item()
    print('sentence : ', sentence)
    print('predict_value %f:' %predict_value)
    print('labeled : %d' %predict_answer)
    return predict_answer
```

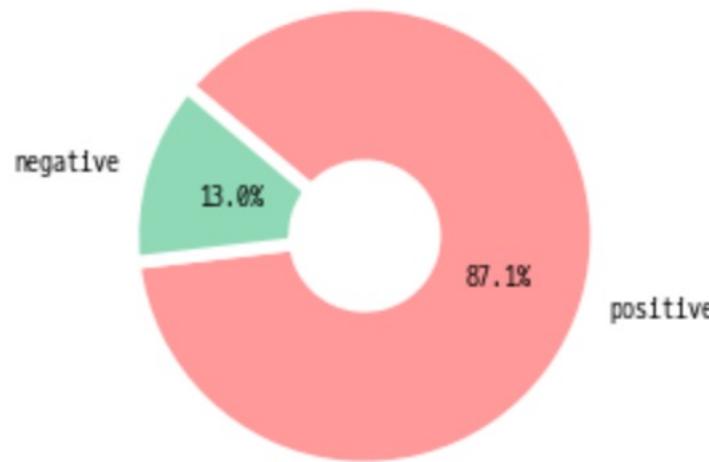
```
for title, item in item_total_data.items():
    label_data = []
    print(title, ' 분류 시작 ----- ')
    for idx in range(len(item)):
        print('idx - %d'%idx)
        label_data.append(review_evaluation_predict(item.iloc[idx]['제목']))
    item['label'] = label_data
```

sentence : 헤리블랙쿠션은 워낙유명한데 이제 처음써보네요이제 마스크는 필수인시대라서 헤리뉴블랙쿠션으로 구입해봤는데 마스크에 묻어남이 완전히 없다고는하진 못할것같아요.그래도 꺽서 뿌려주고 고정해  
predict\_value 0.284912:  
labeled : 0  
idx - 860  
sentence : 이전 블랙쿠션보다 덜 묻어나는 느낌인 것 같아요. 색상은 21호 쓰다가 이번에 21N1으로 구매했는데 원래 홍조가 좀 있는 제 얼굴에도 조금만 많이 바르면 노란끼가 도는 것 같긴  
predict\_value 0.521263:  
labeled : 1  
idx - 861  
sentence : 유튜브 보고 훌린듯이 구매했는데 뭉침없이 지속되는데 너무 좋았고요 마스크 써서 화장 안했는데 이건 묻어남도 거의 없어요ㅠㅠ 얇게 발라도 커버 잘 되고 색도 예뻐요 쿠션이랑 리필만  
predict\_value 0.962805:  
labeled : 1  
idx - 862  
sentence : 누가 쓴 거 받은 줄 알았어요 먼지 쿠션 열자마자 먼지 엄청 불어있고 퍼프 놓는 자리에 찍힘이 두 곳이나 있고 케이스 자체에 기스도 있네요 내용물이 샌 건 아닌데 옆에 하얗게 파  
predict\_value 0.050268:  
labeled : 0  
idx - 863  
sentence : 우선 랜덤 깜빡선을..크..진짜 깜빡스럽네요 개이득ㅎㅎ만족스럽습니당 감사해요5g 증정용 쿠션도 너무 귀엽고 좋아요ㅠ이 사이즈로도 팔면 좋겠어유.. 혹시 5g짜리 리필은 따로 팔진  
predict\_value 0.960615:  
labeled : 1  
idx - 864  
sentence : 매번 홍쇼핑 제품만 쓰다가 처음으로 헤라를 사봤어요~ 색상은 잘 몰라 17nc 21nc 를 샀는데 17nc쪽이 더 맞긴하나~ 21nc도 나쁘지 않아 그냥 때에 맞춰쓰려구요~ nc  
predict\_value 0.695872:  
labeled : 1  
idx - 865  
sentence : 마스크에 잘안묻는다는 후기 보고 너무 사용해보고 싶었어요! 샘플 구성이 너무 일차네요 꺽서까지 오는 줄 몰랐어서 받아보고 놀랐어요ㅎㅎ 피부가 많이 민감한 편이어서 화장품 함부로  
predict\_value 0.956267:  
labeled : 1  
idx - 866  
sentence : 사은품이 이렇게 많다니!!!감동 감사합니다♥잘쓸게요♥'파우치는 못받겠지'했는데 파우치도 오고..사은품으로 주신 쿠션은 리필까지 챙겨주실줄 몰랐어요 진짜👍▶배송도 빠르고 포장도  
predict\_value 0.929520:  
labeled : 1  
idx - 867

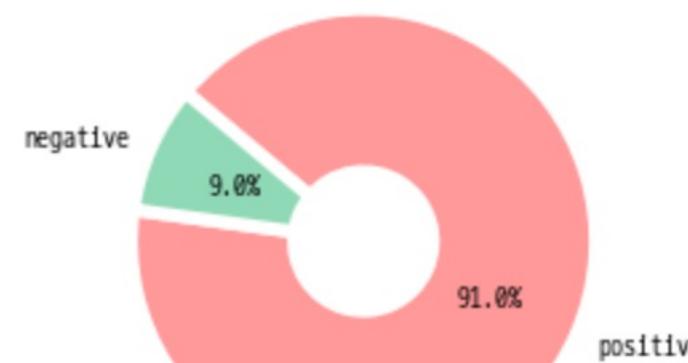


- 감성 분석 결과 도출

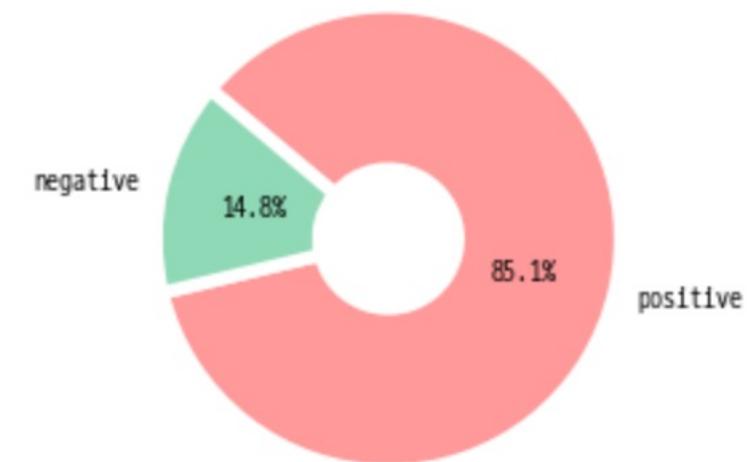
헤라 블랙 쿠션 분류 비교



닥터지 마일드 선크림 분류 비교



삼성 갤럭시 버즈 분류 비교



Graph Ranking 알고리즘 사용하여 중요 키워드 단어 추출 (HITS Algorithm)

STEP 4.

- 감성 분석 결과 도출

해라 블랙 쿠션 긍정 데이터 분석  
scan vocabs ...  
num vocabs = 4395  
done = 10 Early stopped.  
리뉴얼: 16.1948  
마스크: 12.0032  
커버: 9.8397  
피부: 8.3105  
밀착력: 7.0872  
바로: 6.0683  
제품: 5.4096  
조금: 5.3995  
물어: 5.3202  
아직: 5.0770  
아주: 5.0192  
케이스: 4.9878  
얇게: 4.7820  
지속력: 4.5988  
배송: 4.2888  
색상: 4.2606  
21: 4.2459  
실广大群众: 3.4978  
매트: 3.4691  
써보: 3.4023  
구성: 3.1230  
한번: 3.0705  
건조: 3.0591  
주문: 3.0219  
있어: 2.9189  
라이브: 2.8533  
들어: 2.4563  
밝은: 2.4555  
퍼프: 2.3506  
기초: 2.2572

해라 블랙 쿠션 부정 데이터 분석  
scan vocabs ...  
num vocabs = 905  
done = 10  
마스크: 7.4208  
리뉴얼: 3.3028  
물어: 3.1659  
피부: 2.2694  
제품: 2.2518  
그냥: 2.0054  
커버: 1.9561  
그래서: 1.7911  
얇게: 1.7142  
바르고: 1.5918  
색상: 1.5695  
그런지: 1.5553  
주민: 1.5521  
밀착력: 1.5100  
안문: 1.4517  
배송: 1.3887  
조금: 1.3680  
두껍게: 1.3331  
이렇게: 1.3189  
건조: 1.3039  
같은: 1.2980  
21호: 1.2928  
시간: 1.2419  
써도: 1.2379  
했는데: 1.2292  
리필: 1.2247  
하지만: 1.2043  
케이스: 1.1911  
상품: 1.1577  
쓰는데: 1.1479

닥터지 마일드 선크림 긍정 데이터 분석  
scan vocabs ...  
num vocabs = 2093  
done = 10  
백탁: 6.3285  
배송: 5.4336  
발림성: 4.7834  
피부: 4.5408  
저렴하게: 4.4167  
썬크림: 4.2619  
순하고: 3.7015  
없고: 3.6648  
아주: 3.5344  
부드럽게: 3.4188  
아직: 3.3475  
촉촉: 3.3351  
무기자차: 3.2773  
없어서: 3.1168  
계속: 3.0438  
빠르고: 3.0014  
좋습니다: 2.8492  
않아서: 2.7589  
같아요: 2.7341  
순해서: 2.5566  
트러블: 2.4780  
이것만: 2.4412  
적당히: 2.4257  
건조: 2.3865  
최고: 2.2738  
써보고: 2.2526  
있어요: 2.2497  
조금: 2.2438  
성분: 2.1672  
있는: 2.1619

닥터지 마일드 선크림 부정 데이터 분석  
scan vocabs ...  
num vocabs = 328  
done = 10  
백탁: 2.5008  
무기자차: 2.2098  
트러블: 1.8078  
없어요: 1.7062  
이거: 1.5705  
피부: 1.3373  
좋아서: 1.2993  
같아요: 1.2979  
배송: 1.2327  
건조: 1.1721  
발림: 1.0174  
심하: 1.0098  
성분: 0.9954  
얼굴이: 0.9785  
눈이: 0.9386  
시간: 0.9291  
순하: 0.9140  
보통: 0.8940  
주문했: 0.8542  
조금: 0.8532  
있어요: 0.8514  
자외선: 0.8328  
없는: 0.8223  
...: 0.7325  
기름: 0.6812  
상품: 0.6721  
건성: 0.6566  
그럼: 0.6305  
유분: 0.6260  
그렇: 0.6249

삼성 갤럭시 버즈 긍정 데이터 분석  
scan vocabs ...  
num vocabs = 2719  
done = 10  
배송: 11.5060  
라이브: 7.6024  
아주: 7.1971  
음질: 6.9122  
선물로: 6.7386  
귀에: 6.7322  
귀가: 6.6985  
착용: 6.3051  
빠르고: 6.1284  
오픈형: 5.9770  
케이스: 5.5242  
이어폰: 4.9429  
노이즈: 4.7410  
바로: 4.0117  
브론즈: 3.6388  
무선: 3.4661  
가격: 3.2205  
들어: 3.1915  
이쁘고: 3.0025  
좋습니다: 2.9807  
커널형: 2.9660  
에어팟: 2.9437  
색상: 2.8622  
생각: 2.8567  
최고: 2.8147  
블루투스: 2.7472  
예쁘고: 2.6759  
디자인: 2.6442  
저렴하게: 2.6046  
조금: 2.5397

삼성 갤럭시 버즈 부정 데이터 분석  
scan vocabs ...  
num vocabs = 698  
done = 10  
귀가: 3.8218  
귀에: 3.6066  
배송: 3.5346  
착용: 3.2066  
이어폰: 2.9081  
음질: 2.8460  
생각보다: 2.5575  
노이즈: 2.4216  
오픈형: 2.1724  
라이브: 1.9316  
같은: 1.7332  
이건: 1.6859  
작아서: 1.6555  
별로: 1.5585  
있는: 1.4798  
커널형: 1.4784  
이렇게: 1.4538  
따라: 1.4400  
장시간: 1.3761  
매우: 1.3676  
캔슬링: 1.3473  
조금: 1.2698  
문제: 1.2601  
...: 1.2250  
그냥: 1.2249  
상품: 1.2163  
어제: 1.1975  
작은: 1.1951  
주로: 1.1857  
에어팟: 1.1847

01

## 분석 과정 및 방법

중요 키워드 단어로 wordcloud 생성 및 저장

#### STEP 4.

```
krwordrank_cloud = krwordrank_cloud.generate_from_frequencies(keyword)
fig = plt.figure(figsize=(10, 10))
plt.imshow(krwordrank_cloud, interpolation="bilinear")
plt.show()
fig.savefig('./wordcloud_data/삼성갤럭시버즈_negative_wordcloud')
```

### - 감성 분석 결과 도출

- 중요 키워드 추출 및 wordcloud 생성

## 헤라 블랙 쿠션 긍정 키워드



## 헤라 블랙 쿠션 부정 키워드



# 평가 및 결과

실제 네이버 리뷰 데이터로 BertClassificationModel 학습 후 평가 결과

정확도 - 0.89

```
WARNING:CONSOLE: The parameter `return_dict` cannot be set in graph mode and will always be set to `true`.  
313/313 [=====] - 390s 1s/step - loss: 0.6157 - accuracy: 0.6397 - val_loss: 0.3573 - val_ac  
curacy: 0.8630  
Epoch 2/2  
313/313 [=====] - 338s 1s/step - loss: 0.3177 - accuracy: 0.8799 - val_loss: 0.3007 - val_ac  
curacy: 0.8900
```

```
from sklearn.metrics import classification_report  
y_true = test['label']  
# F1 Score 확인  
print(classification_report(y_true, np.round(preds,0)))
```

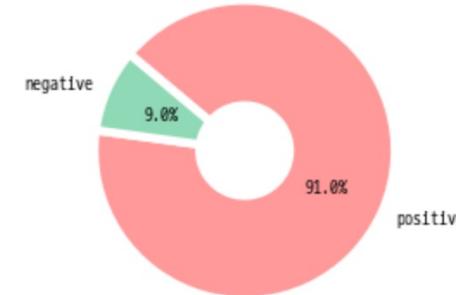
	precision	recall	f1-score	support
0	0.94	0.83	0.88	4991
1	0.85	0.95	0.90	5004
accuracy			0.89	9995
macro avg	0.90	0.89	0.89	9995
weighted avg	0.90	0.89	0.89	9995



닥터지 마일드 선크림 분류 비교

## 닥터지 마일드 선크림 실제 리뷰 분석 결과

- : 배송이 빠르다, 발림성이 좋다, 순하다, 무기자차, 등에 관한 긍정적인 내용과
- : 트러블이 있었다, 백탁현상이 심하다, 건조하다, 눈이 시리다, 안맞다는 부정적인 내용이 도출됨



닥터지 마일드 선크림 긍정 데이터 분석

scan vocabs ...

num vocabs = 2093

done = 10

백탁:	6.3285	백탁:	2.5008
배송:	5.4336	무기자차:	2.2098
발림성:	4.7834	트러블:	1.8078
피부:	4.5408	없어요:	1.7062
저렴하게:	4.4167	이거:	1.5705
썬크림:	4.2619	피부:	1.3373
순하고:	3.7015	좋아서:	1.2993
없고:	3.6648	같아요:	1.2979
아주:	3.5344	배송:	1.2327
부드럽게:	3.4188	건조:	1.1721
아직:	3.3475	발림:	1.0174
촉촉:	3.3351	심하:	1.0098
무기자차:	3.2773	성분:	0.9954
없어서:	3.1168	얼굴이:	0.9785
계속:	3.0438	눈이:	0.9386
빠르고:	3.0014	시간:	0.9291
좋습니다:	2.8492	순하:	0.9140
않아서:	2.7589	보통:	0.8940
같아요:	2.7341	주문했:	0.8542
순해서:	2.5566	조금:	0.8532
트러블:	2.4780	있어요:	0.8514
이것만:	2.4412	자외선:	0.8328
적당히:	2.4257	없는:	0.8223
건조:	2.3865	...:	0.7325
최고:	2.2738	기름:	0.6812
써보고:	2.2526	상품:	0.6721
있어요:	2.2497	건성:	0.6566
조금:	2.2438	그런:	0.6305
성분:	2.1672	유분:	0.6260
있는:	2.1619	그렇:	0.6249

닥터지 마일드 선크림 부정 데이터 분석

scan vocabs ...

num vocabs = 328

done = 10

## 닥터지 마일드 선크림 긍정 키워드



## 닥터지 마일드 선크림 부정 키워드





## 아쉬운 점

- 네이버 쇼핑몰 리뷰 데이터 수집이 최대 2000개만 제공함 (실제 데이터 갯수 10000개 이상)
- 감성 분류 모델의 정확도가 0.89로 엄청 높지만 문장의 길이가 너무 길면 제대로 분류가 안되는 경우 존재함
- 감성 분류 후, 키워드 추출 결과 긍정과 부정의 키워드가 모호한 경우 존재함

## 참고 자료

koBERT (한국어 자연어 처리기) - <https://github.com/SKTBrain/KoBERT>

오픈 소스 데이터 - 네이버 쇼핑 제품별 리뷰 데이터 <https://github.com/bab2min/corpus/tree/master/sentiment>

크롤링 참고 - <https://github.com/jinho9613/NaverShoppingReviewCrawler/blob/master/NaverShoppingCrawler.py>

HuggingFace 오픈소스 라이브러리 - <https://huggingface.co/transformers/>

키워드 추출 라이브러리 오픈소스 - <https://github.com/lovit/KR-WordRank>

## 분석 툴



PYTHON



감사합니다 !

