



멀티캠퍼스 세미 프로젝트 : 데이터 구조

한국인의 밥상 준비

/ CONTENTS /

01

[프로젝트 개요](#)

- 기획 배경
- 최종 목표
- 구성원 및 역할

02

[프로젝트 수행 절차 및 방법](#)

- 선정 데이터 소개
- 데이터 전처리 과정
- 키워드 추출 과정
- 시각화 과정

03

[기대 효과](#)

- 참고 키워드 제시
- 자가점검 가능한 체크리스트

04

[회고](#)

- KPT 회고

01

프로젝트 개요



퇴직 후 고령층의
일자리 부족

고령화 지속
MZ 세대들의 일자리 기피





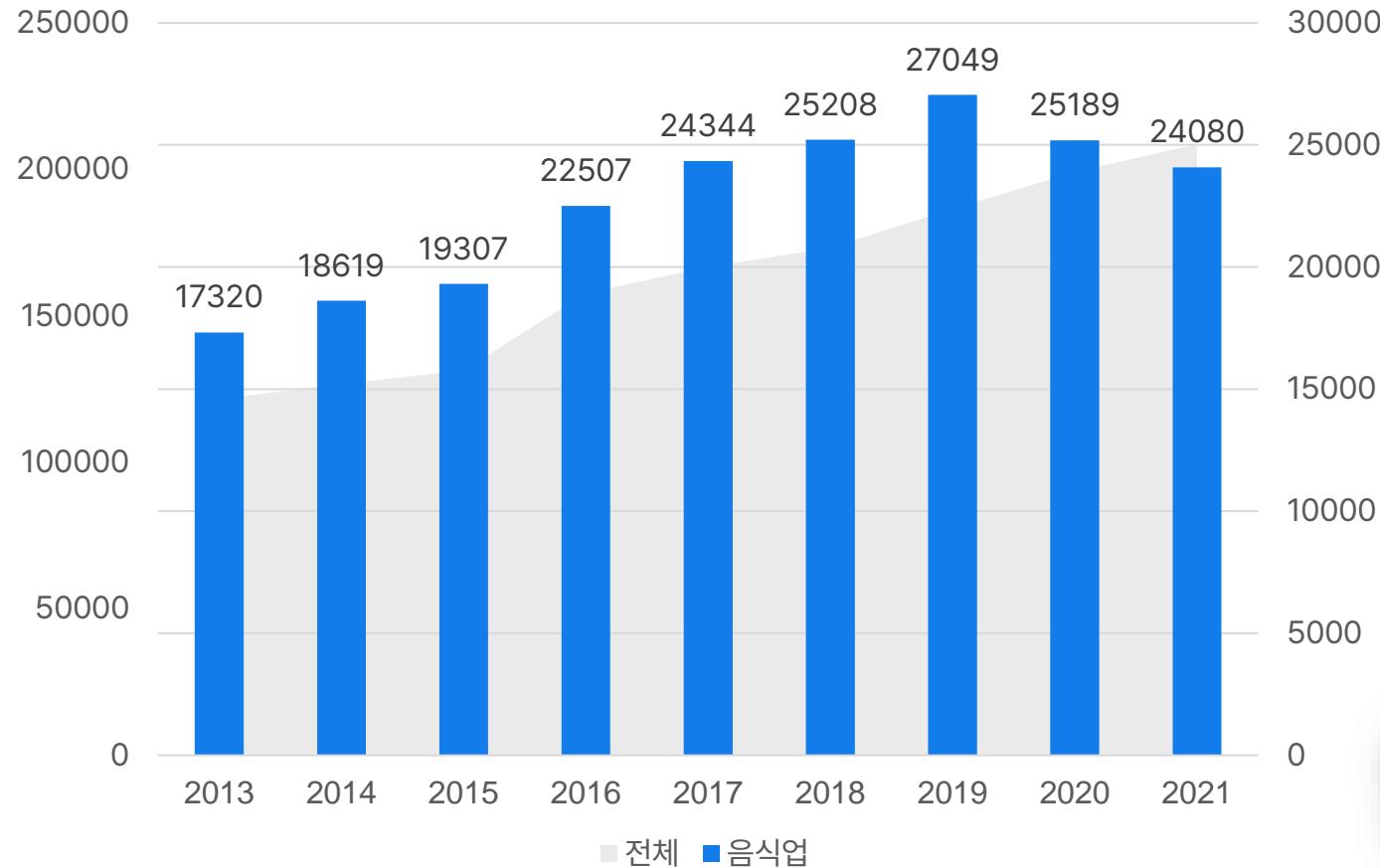
퇴직 후 고령층의
일자리 부족

“생계를 위한 창업 시작”

고령화 지속

MZ 세대들의 일자리 기피

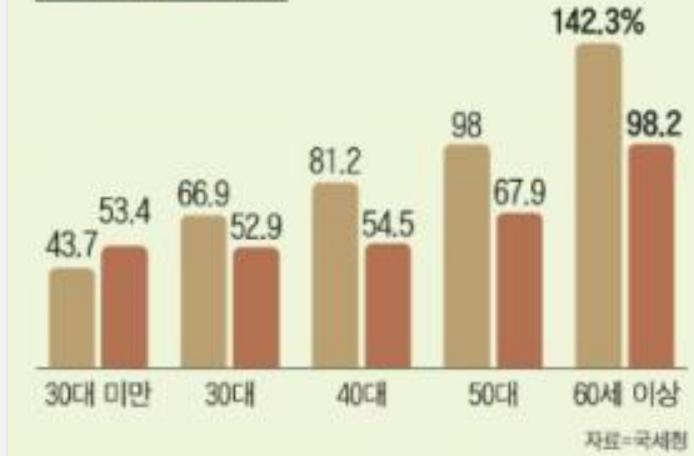
60세 이상 폐업 추이



© 국세청

연령대별 창업자 대비 폐업자 비율 2021년 기준

■ 음식업 ■ 소매업



© 조선일보



60세 이상 100명이 음식점 낼때, 142명이 망한다



한겨레
연금 받는 노인 절반, 일 못 놓는다... '나홀로 창업'도 증가

— 나고 마이 라이프

지난해 고령자 취·창업자 역대 최다... 양질 일자리는 아직



고령층의 요식업 창업 중 비중이 높은 한식당 리뷰를 분석하여,

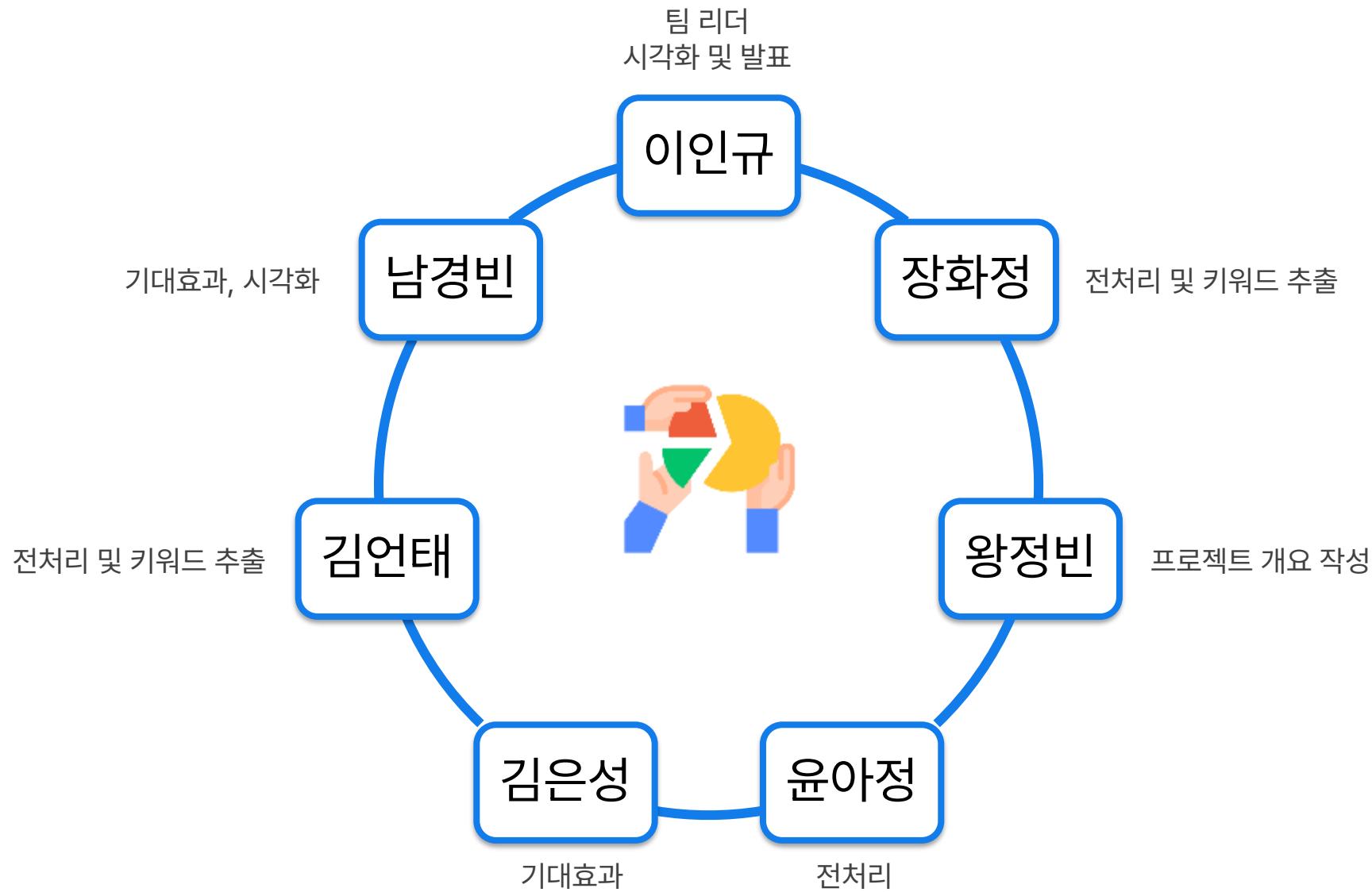
창업 전 또는 운영에 있어 확인할 수 있는

데이터의 시각화를 통해 주요 키워드와, 체크리스트를 제공함으로써

고령층 식당 폐업률 감소에 기여한다.



구성원 및 역할



02

프로젝트 수행 절차 및 방법

KR3: Korean Restaurant Reviews with Ratings

460K(+180K) Korean Sentiment Classification Dataset

© Kaggle

선정 사유 .

- 고령층의 창업 분야 중 요식업, 특히 한식당이 큰 비중을 차지하고 있는 상황
- ∴ 한식당에 대한 실질적 고객 리뷰를 통해 폐업률 감소에 기여하고자 본 데이터셋을 선정

데이터 특징 .

- Rating 컬럼은 0(부정), 1(긍정), 2(모호함)으로 분류되어 있는 감정 분석을 위한 데이터셋
- 구조: 4개의 컬럼 x 60만개 이상의 행

```
1 ### Raw 데이터 로드
2 df = pd.read_csv('kr3_raw.tsv', sep = '\t', encoding = 'utf-8')
3 df
```

	Region	Rating	Category	Review
0	인덕원	1	삼겹살 고기집	숙성 돼지고기 전문점입니다. 건물 모양 때문에 매장 모양도 좀 특이하지만 쾌적한 편...
1	인덕원	1	삼겹살 고기집	고기가 정말 맛있었어요! 육즙이 가득 있어서 너무 좋았아요₩n일하시는분들 너무 친절하...
2	인덕원	1	순대국 찹쌀순대	잡내없고 깔끔, 담백한 맛의 순대국이 순대국을 안 좋아하는 사람들에게도 술술 넘어갈...
3	인덕원	1	순대국 찹쌀순대	고기 양이 푸짐해서 특 순대국밥을 시킨 기분이 듭니다~~ 맛도 좋습니다₩n다만 양념...
4	인덕원	1	순대국 찹쌀순대	순대국 자체는 제가 먹어본 순대국밥집 중에서 Top5 안에는 들어요₩n그러나 ...
...
642408	평택	2	일식/중식/세계음식	요즘, 핫하게,,떠오르구 있는 중국집.ㅋ, 맥주의 여파로 속이 안좋지만 와콤...
642409	평택	0	한식	원래 글 안쓰는데 이거는 정말 다른분들 위해서 써야할것같네요 방금 포장주문 해서 왔...
642410	평택	1	한식	우리팀 단골집, 술먹고 다음 날 가면 푸짐하게 배불리 해장 할 수 있는곳, 주말도 ...
642411	평택	2	카페/디저트	원래는 평택에 있었는데, 연남동에도 최근에 생겨서 방문 했는데..진짜 줄이 어마어마...
642412	평택	2	한식	친구들의 추천으로 가보게 된 곳. 안성과 평택 몇군데 위주로 체인점이 있는 소규모 ...
642413 rows × 4 columns				

① 결측치 제거

② Rating의 '2(모호함)'을 가진 행 제거

③ Category 컬럼에서 한식에 포함되는 행 추출



```
1 ### 데이터 전처리
2 # 누락 데이터 확인
3 df.isnull().sum()
4
5 # 결측치 제거 및 제거 확인
6 df = df.dropna()
7 df.isnull().sum()
```

Region 0
Rating 0
Category 0
Review 0
dtype: int64

```
1 ### Rating 컬럼에서 '2(모호함)'을 가진 행 제거
2 df = df[df['Rating'] != 2]
3 df = df.reset_index()
4 df.tail()
```

	index	Region	Rating	Category
459522	642404	평택	0	일식/중식/세계음식
459523	642405	평택	1	한식 송탄
459524	642406	평택	1	한식 집에
459525	642409	평택	0	한식 원래

```
1 ### 지정한 카테고리, 한식 관련 리뷰만 추출
2 df_category = df.loc[(df['Category'].isin(['한식', '기타 한식',
3 '삼겹살', '고기/구이류', '고기 요리', '탕/찌개/전골',
4 '한정식/백반/전통 한식', '갈비', '한정식', '', '냉면']),
5 ['Rating', 'Review'])].reset_index(drop=True)
```

	Rating	Review
0	1	개인적으로 제일 좋아하는 냉면집 진짜 집 근처였으면 일주일에 한번씩 같았어요
1	1	방화동 구석의 정말 허름한 노포. 아저씨의 응대가 유쾌하고 살갑다. 손으로 쓴 메뉴...
2	1	아주머니 혼자하시는 식당이고 푸짐해요 국물이 사골이라 맛있어요 순대는 쓰쓰해요
3	1	장어가 담백하면서 고소하고 맛있습니다. 비린맛이 적어 물리지 않고 맛있게 먹을 수 ...
4	1	오픈 이벤트로 막국수 할인 중이며, 맛있고 푸짐하여 가족 외식으로 괜찮은 곳

④ 특수, 개행(\n), 이모지 제거 · 정규화 ⑤ hanspell 라이브러리로 맞춤법 검사 실시 ⑥ 이모지 및 특수문자 재제거

```
1 # 필요 라이브러리 임포트
2 from hanspell import spell_checker

1 # 맞춤법 검사 함수 생성
2 def spell_check(review):
3     result = spell_checker.check(review)
4     return result.checked

1 # Review 열의 데이터 타입 변경
2 df_category['Review']=df_category['Review'].astype('str')
3
4 # 맞춤법 검사기 실시
5 df_category['review_spell_check'] = df_category['Review'].apply(spell_check)
```

```
1 # 이모지 및 특수문자 제거
2 import re
3
4 def remove_special_characters(text):
5     pattern = r'[^\\w\\sㄱ-ㅣ가-힣]' # 특수문자나 이모지 패턴
6     clean_text = re.sub(pattern, '', text)
7     return clean_text
8
9 df['clean_Review'] = df['Review'].apply(remove_special_characters)
10 df['clean_review_spell_check'] = df['review_spell_check'].apply(remove_special_characters)
11
```



키워드 추출 과정과 목표

- ① 전처리한 데이터에서 새로운 긍정, 부정 키워드를 선택!
- ② 키워드가 포함되면 1, 아니면 0 을 매핑하여 총점 점수화
- ③ 긍정, 부정 리뷰를 새로 판단!

[상인들에게 도움이 되는 키워드 추출해보자!]

▼ 전처리된 데이터

index	Rating	Review	review_spell_check
0	1	개인적으로 제일 좋아하는 냉면집진짜 집 근처였으면 일주일에 한번씩 갈것같아요	개인적으로 제일 좋아하는 냉면집 진짜 집 근처였으면 일주일에 한 번씩 갈 것 같아 요
1	1	방화동 구석의 정말 허름한 노포 아저씨의 응대가 유쾌하고 살갑다 손으로 쓴 메뉴판 본게 대체 언제인지 맛있게 한끼 잘 먹을 수 있는 좋은 집	방화동 구석의 정말 허름한 노포 아저씨의 응대가 유쾌하고 살갑다 손으로 쓴 메뉴판 본게 대체 언제인지 맛있게 한끼 잘 먹을 수 있는 좋은 집
2	1	아주머니 혼자하시는 식당이고 푸짐해요국물이 사골이라 맛있어요 순대는 쏘쏘해 요	아주머니 혼자 하시는 식당이고 푸짐해요 국물이 사골이라 맛있어요 순대는 쏘쏘해 요
3	1	장어가 담백하면서 고소하고 맛있습니다 비린맛이 적어 물리지 않고 맛 있게 먹을 수 있어요	장어가 담백하면서 고소하고 맛있습니다 비린 맛이 적어 물리지 않고 맛있게 먹을 수 있어요
4	1	오픈 이벤트로 막국수 할인 중이며 맛있고 푸짐하여 가족 외식으로 괜찮은 곳	오픈 이벤트로 막국수 할인 중이며 맛있고 푸짐하여 가족 외식으로 괜찮은 곳
5	1	삼겹살 고기랑 지방의 비율 굿고기질좋고 고추마늘팍팍주시니김치가 신스틸러	삼겹살 고기랑 지방의 비율 굿 고기 질 좋고 고추 마늘 팍팍 주시니 김치가 신 스틸러
6	1	맛있어요 점심에 가면 사람 많아서 조금 기다려야 할 수 도 있어요	맛있어요 점심에 가면 사람 많아서 조금 기다려야 할 수도 있어요
7	1	냉면이 매울 때 만두를 사이드로 같이 먹으면 맛있어요	냉면이 매울 때 만두를 사이드로 같이 먹으면 맛있어요
8	1	깔끔해서 정말 좋은곳 위생점수 백만점 맛도 맛있고 친절해서 좋다	깔끔해서 정말 좋은 곳 위생 점수 백만 점 맛도 맛있고 친절해서 좋다
9	1	무난하게맛있어요비냉이 맛있고냉면 사리 추가로 드시면 4000원 입니다 굳이 9천 원 내지 마세요	무난하게 맛있어요 비냉이 맛있고 냉면 사리 추가로 드시면 4000원입니다 굳이 9천 원 내지 마세요
10	1	워낙 퓨전요리 좋아하긴하는데 여기 정말 맛있네요 반주용으로 만들어서 그런지 조 금 간이 세긴한데 그게 더 맛있게 느껴지는듯 합니다	워낙 퓨전요리 좋아하긴하는데 여기 정말 맛있네요 반주용으로 만들어서 그런지 조 금 간이 세긴 한데 그게 더 맛있게 느껴지는듯합니다

54366 rows × 3 columns

새로운 긍정, 부정 키워드 선택하기

[Step 1.] 선정한 10개의 키워드

```
df[ '맛있어요' ] = 0    df[ '맛없어요' ] = 0
df[ '친절' ] = 0        df[ '불친절' ] = 0
df[ '깔끔' ] = 0        df[ '최악' ] = 0
df[ '최고' ] = 0        df[ '별로' ] = 0
df[ '가성비' ] = 0      df[ '다신' ] = 0
```

[Step 2.] '맛있'이라는 키워드로 전체 데이터를 검색 → 애매하게 사용된 '맛있' 키워드 제거!

```
positive_list = ['맛있지는', '감칠맛있개', '맛있는진', '아주맛있지도', '맛있는녀석들', '맛있는녀석들도',
'애매한맛위가맛있는건지', '맛있는지는', '맛있다는건지', '맛있는건가', '감칠맛있고', '맛있을줄', '맛있는맛은',
'맛있을텐데', '맛있을겁니다', '맛있던곳인데', '맛있다는건가', '맛있지는', '안맛있는게', '맛있다고해서', '맛있어서웨이팅길겠거니그냥고',
'맛있다고하길래', '맛있는거였나', '맛있다하기엔', '맛있으려나', '맛있냐', '맛있었던건지', '맛있어지는파리겨울엔', '맛있었음근데자금은',
'그닥맛있게', '맛있어봤자얼마나', '맛있다는건아이고김밥빼고', '맛있어야하는데밥이질다두번이용은', '작년5월에댕겨웠는데진짜의문인곳뭐가맛있다는건지',
'더맛있는곳가세요', '맛있다기보단', '맛있었는데요즘은', '맛있었는데', '맛있진않다떡볶이는',
'맛있음http://mblog.naver.com/PostView.nhn?blogId=yujini03101&logNo=220402717533', '맛있진않아도', '여기왜맛있다는건지다들', '맛있었는데풀면은',
'하려가고싶은맛맛있계먹은사람이', '맛있을리가', '망망맛있다고', '오이소박이추가주문까지시켜먹었는데그맛있는그집이아니었습니다너무달아요',
'맛있음거라하던데', '맛있게쁘게아니였네요', '안맛있음', '맛있다고해서', '맛있진않아도', '맛있는진', '맛있는걸까요', '그렇게맛있다는데',
```

[Step 3.] 전체 데이터에서 긍정적인 리뷰가 아닌 리뷰는 제외, '맛있어요' 컬럼에 1을 매핑

```
exclusion_patterns1 = [word for word in positive_list]
exclusion_regex1 = '|'.join(exclusion_patterns1)
df.loc[df['review_spell_check'].str.contains('맛있', na=False) & ~df['review_spell_check'].str.contains(exclusion_regex1, na=False),
'맛있어요' ] = 1
```

[Step 4.] '친절' 키워드를 찾기위해 '친절' 키워드로 전체 데이터를 검색 부정적 리뷰인 '불친절'을 제외,
'친절'이라는 키워드 존재 시 → '친절' 컬럼에 1을 매핑

```
df.loc[df['review_spell_check'].str.contains('친절', na=False) & ~df['review_spell_check'].str.contains('불친절', na=False), '친절'] = 1
```

```
df.loc[df['review_spell_check'].str.contains('깔끔', na=False), '깔끔'] = 1
```

```
df.loc[df['review_spell_check'].str.contains('최고', na=False), '최고'] = 1
```

같은 방식으로 '깔끔', '최고' 키워드를 전체 데이터에서 찾아 1을 매핑

'맛있', '친절', '깔끔', '최고'가 포함되지 않은 리뷰에는 0으로 매핑!

```
negative_del_list = ['누가맛없데', '누가맛없데', '맛없는게없어요', '입맛없었는데', '아주맛없나하면', '맛없어짐살은', '맛없다고하면', '맛없다는',
                     '맛없을수가', '맛없는', '맛없지도', '맛없는', '맛없기로', '맛없는게없더라구요', '입맛없을', '입맛없을', '입맛없을때가면', '맛없다고',
                     '백숙맛없을줄알았는데', '입맛없을때', '맛없다고는', '입맛없을때', '김은맛없음', '입맛없을때', '매운맛없이', '맛없을거같았는데', '단맛없는',
                     '맛없을수가없지', '맛없진않은데', '조미료맛없이', '비린맛없는', '맛없는건아닌데', '맛없는게없네요', '조미료맛없이', '맛없어서']
exclusion_patterns2 = [word for word in negative_del_list]

exclusion_regex2 = '|'.join(exclusion_patterns2)
df.loc[df['review_spell_check'].str.contains('맛없', na=False) & ~df['review_spell_check'].str.contains(exclusion_regex2, na=False),
       '맛없어요'] = 1
```

```
df.loc[df['review_spell_check'].str.contains('불친절', na=False), '불친절'] = 1

df.loc[df['review_spell_check'].str.contains('최악', na=False), '최악'] = 1

df.loc[df['review_spell_check'].str.contains('별로', na=False), '별로'] = 1

df.loc[df['review_spell_check'].str.contains('다신', na=False) | df['review_spell_check'].str.contains('다시는', na=False), '다신'] = 1
```

```
df['pos_total'] = df[['맛있어요', '친절', '깔끔', '최고']].sum(axis=1)
```

```
df['neg_total'] = df[['맛없어요', '불친절', '최악', '별로', '다신']].sum(axis=1)
```



키워드 추출 결과 및 방향 전환

- ① 키워드의 총점으로는 분석이 어렵다고 판단
- ② MeCab 형태소 분석기를 사용 → 빈도수 분석!
- ③ 큰 틀의 키워드를 선택 후 세부 키워드 추출!

▼ MeCab을 활용한 형태소 분석 수행

[Step 1.] MeCab 설치

```
### konlpy, Mecab 형태소 분석기 설치 스크립트 실행 - 설치하는데 2분 10초 ~ 20초
!curl -s https://raw.githubusercontent.com/teddylee777/machine-learning/master/99-M
# 파이썬에서 mecab을 사용하기 위한 설치
!pip install python-mecab-ko
```

[Step 3.] 2차원 리스트 → 1차원 리스트로 변환

```
# 2차원 리스트인 tmp를 1차원 리스트로 바꾸기
import itertools
list1 = list(itertools.chain(*tmp))
```

[Step 2.] MeCab을 활용한 명사 추출

```
# MeCab을 활용한 명사 추출
from mecab import MeCab
mecab = MeCab()

tmp = []
for sentence in df['review_spell_check']:
    tmp.append(mecab.nouns(sentence))
```

[Step 4.] 빈도수 순으로 10,000개의 단어 추출

```
# 빈도수 순으로 10000개 추출
from collections import Counter
print(Counter(list1).most_common(10000))
```

▼ MeCab 형태소 분석기를 활용하여 빈도수 순 단어 출력 결과

빈도수 순으로 10000개 추출

```
from collections import Counter
print(Counter(list1).most_common(10000))
```

(('맛', 17271), ('고기', 8059), ('집', 7704), ('곳', 5982), ('가격', 5516), ('것', 5379), ('거', 4967), ('맛집', 4897), ('수', 4790), ('음식', 4496), ('친절', 4371), ('여기', 4102), ('때', 3857), ('양', 3725), ('밥', 3643), ('원', 3544), ('반찬', 3459), ('듯', 3396), ('최고', 3305), ('사람', 3126), ('개', 3119), ('번', 3000), ('분', 2989), ('방문', 2954), ('만', 2905), ('냉면', 2904), ('추천', 2829), ('서비스', 2743), ('직원', 2673), ('정도', 2567), ('떡볶이', 2471), ('생각', 2436), ('양념', 2363), ('국물', 2268), ('갈비', 2258), ('메뉴', 2200), ('식당', 2182), ('하나', 2131), ('중', 2045), ('데', 2032), ('사장', 1993), ('시간', 1976), ('주문', 1933), ('족발', 1861), ('식사', 1798), ('만족', 1768), ('분위기', 1767), ('점', 1709), ('말', 1689), ('곱창', 1681), ('나', 1606), ('인분', 1592), ('개', 1584), ('전', 1515), ('느낌', 1499), ('만두', 1497), ('다음', 1486), ('김치', 1481), ('웨이팅', 1469), ('편', 1432), ('기분', 1422), ('친구', 1386), ('내', 1361), ('가족', 1327), ('손', 1323), ('삼겹살', 1323), ('김밥', 1322), ('줄', 1275), ('육수', 1274), ('볶음밥', 1269), ('천', 1267), ('불친절', 1261), ('밀반찬', 1259), ('저', 1257), ('건', 1219), ('돈', 1209), ('일', 1207), ('기대', 1197), ('시', 1193), ('육회', 1175), ('굿', 1166), ('간', 1137), ('정식', 1127), ('우럭', 1107), ('기본', 1098), ('떡갈비', 1093), ('명', 1093), ('점심', 1066), ('가게', 1066), ('튀김', 1060), ('인', 1050), ('돼지', 1040), ('추가', 1034), ('제주', 1034), ('년', 1025), ('감추', 1010), ('처음', 1002), ('제', 995), ('비빔밥', 994), ('돈가스', 993), ('가성', 985), ('소스', 972), ('테이블', 970), ('물', 967), ('이상', 961), ('근처', 945), ('입', 945), ('예약', 944), ('자리', 927), ('평양냉면', 924), ('한우', 908), ('대비', 898), ('입맛', 892), ('배', 888), ('구이', 882), ('세트', 877), ('걸', 873), ('개인', 871), ('면', 863), ('된장찌개', 846), ('포장', 844), ('냄새', 837), ('고깃집', 833), ('불고기', 827), ('칼국수', 822), ('치즈', 820), ('조림', 812), ('개장', 807), ('풀', 795), ('부모', 770), ('앞', 764), ('최악', 758), ('날', 750), ('때문', 749), ('삼', 746), ('목살', 743), ('서울', 731), ('주차', 729), ('리뷰', 728), ('대기', 716), ('막국수', 710), ('소고기', 705), ('떡', 696), ('저녁', 676), ('여행', 665), ('사진', 661), ('살', 660), ('이', 660), ('손님', 656), ('순대', 650), ('대박', 642), ('실망', 641), ('듬', 632), ('갈비탕', 630), ('예전', 628), ('술', 627), ('재료', 624), ('안', 618), ('그릇', 612), ('동네', 609), ('비', 603), ('회', 600), ('간장', 598), ('국수', 597), ('질', 596), ('속', 588), ('인생', 588), ('남', 587), ('찜', 586), ('가능', 585), ('후', 584), ('성비', 582), ('육즙', 577), ('보쌈', 567), ('여긴', 567), ('필요', 555), ('오랜만', 553), ('오', 552), ('아이', 551), ('닭', 550), ('식감', 546), ('요리', 544), ('위생', 543), ('수육', 543), ('건지', 529), ('이곳', 520), ('김치찌개', 519), ('돼지갈비', 518), ('장사', 517), ('존', 517), ('쌈', 515), ('옆', 513), ('서빙', 512), ('가지', 508), ('향', 507), ('이유', 506), ('자극', 501), ('짱', 498), ('국밥', 493), ('대', 493), ('겁', 489), ('보통', 489), ('생선', 486), ('전체', 486), ('건강', 484), ('특별', 476), ('만큼', 473), ('야채', 471), ('반', 470), ('어묵', 470), ('리필', 467), ('기억', 454), ('평양', 452), ('무엇', 452), ('마지막', 442), ('갈비찜', 439), ('코스', 439), ('무침', 438), ('소주', 436), ('라면', 433), ('다양', 429), ('우리', 428), ('차', 423), ('자체', 422), ('돼지고기', 419), ('메밀', 419), ('여자', 417), ('끼', 416), ('막창', 414), ('꼴', 405), ('일품', 403), ('김', 402),

- 큰 키워드 : 위생
- 작은 키워드 : 담배, 먼지, 벌레, 깔끔, 청결, 환기, 머리카락

```
# 위생 단어 찾기
clean = ['담배', '먼지', '벌레', '깔끔', '청결', '환기', '머리카락']
clean = '|'.join(clean)
df_clean = df.loc[df['review_spell_check'].str.contains(clean, na=False)].drop(['Review', '맛있어요', '친절', '깔끔', '최고',
'가성비', '맛없어요', '불친절', '최악', '별로', '다신', 'pos_total', 'neg_total'], axis=1)
df_clean.head()
```

index	review_spell_check
8	깔끔해서 정말 좋은 곳 위생 점수 백만 점 맛도 맛있고 친절해서 좋다
24	백반이 4천 원이라서 들어가 본 집 함바집 같은 지하 식당이라고 생각했는데 의외로 깨끗한 밥과 밀반찬 무한리필이라고 쓰여있어서 두 번 놀람 김치찌개가 맛있고 깔끔한 반찬 맛에 또 한 번 놀람 여기서 맛은 가성 비 맛을 의미함 결론적으로 저렴한 혼밥 간단한 한끼 하기에 최상의 식당이라고 판단됨
26	간단하게 점심 식사하러 들렸어요 매장 분위기가 깔끔하고 순댓국 맛도 나쁘지 않습니다
35	4층 식당가에 있어서 조용하게 먹기 좋음 깔끔하고 맛있음

```
# 위생 단어 찾기 2
# '담배', '먼지', '벌레', '깔끔', '청결', '환기', '머리카락'
clean = []
for my_data in df_clean.iloc:
    data = (my_data['review_spell_check']).split(' ')
    for i in data:
        if '담배' in i and len(i) == 2:
            clean.append(i)
        elif '먼지' in i and len(i) == 2:
            clean.append(i)
        elif '벌레' in i and len(i) == 2:
            clean.append(i)
        elif '깔끔' in i and len(i) == 2:
            clean.append(i)
        elif '청결' in i and len(i) == 2:
            clean.append(i)
        elif '환기' in i and len(i) == 2:
            clean.append(i)
        elif '머리카락' in i and len(i) == 4:
            clean.append(i)

print(pd.Series(data=clean).value_counts())
```

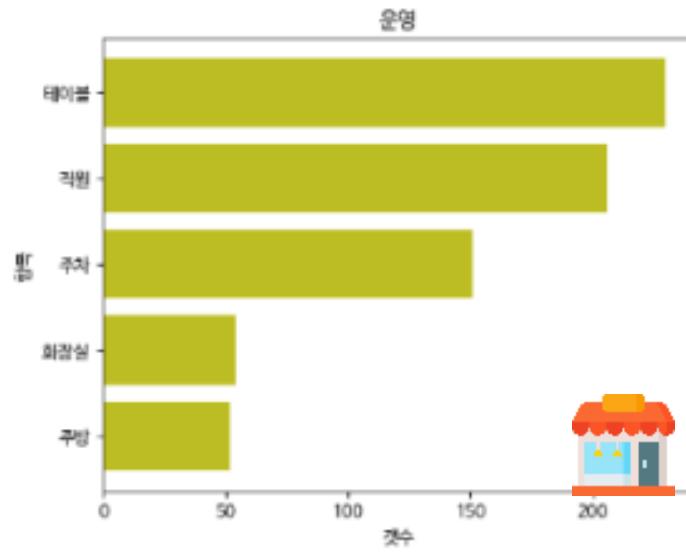
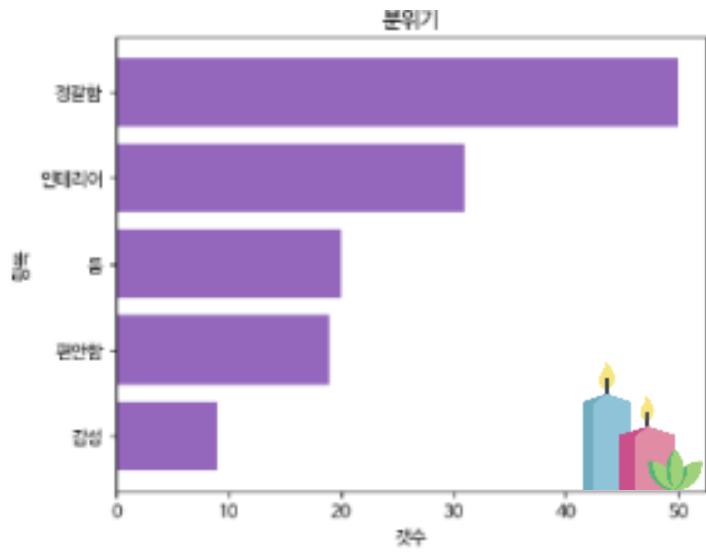
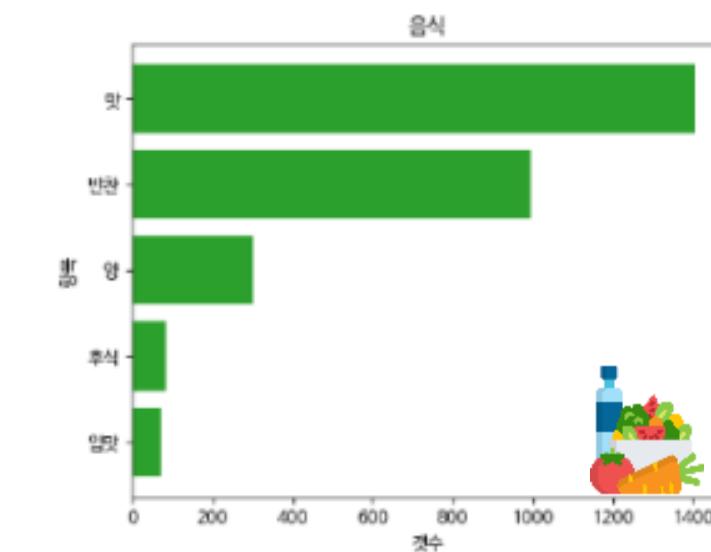
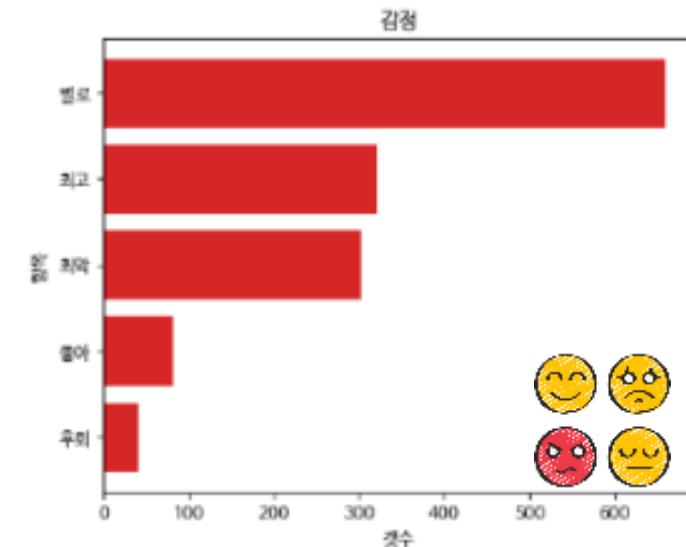
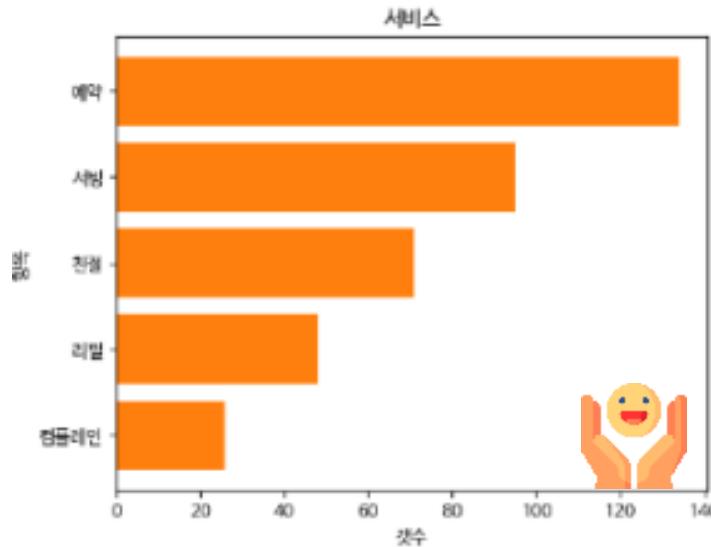
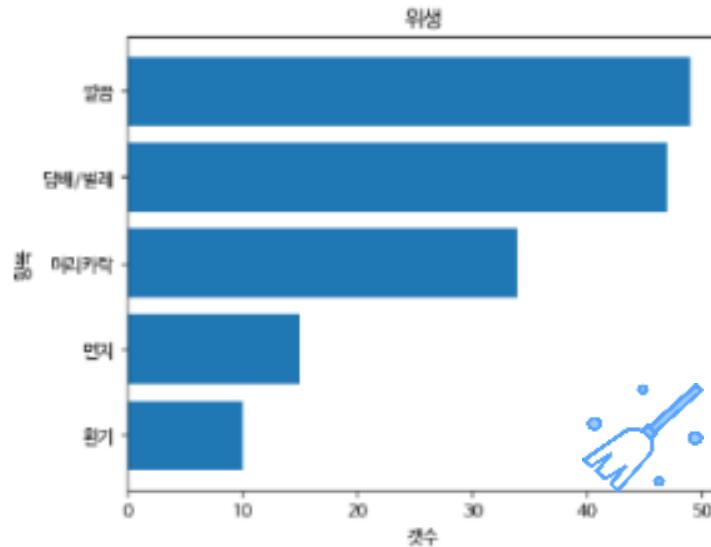


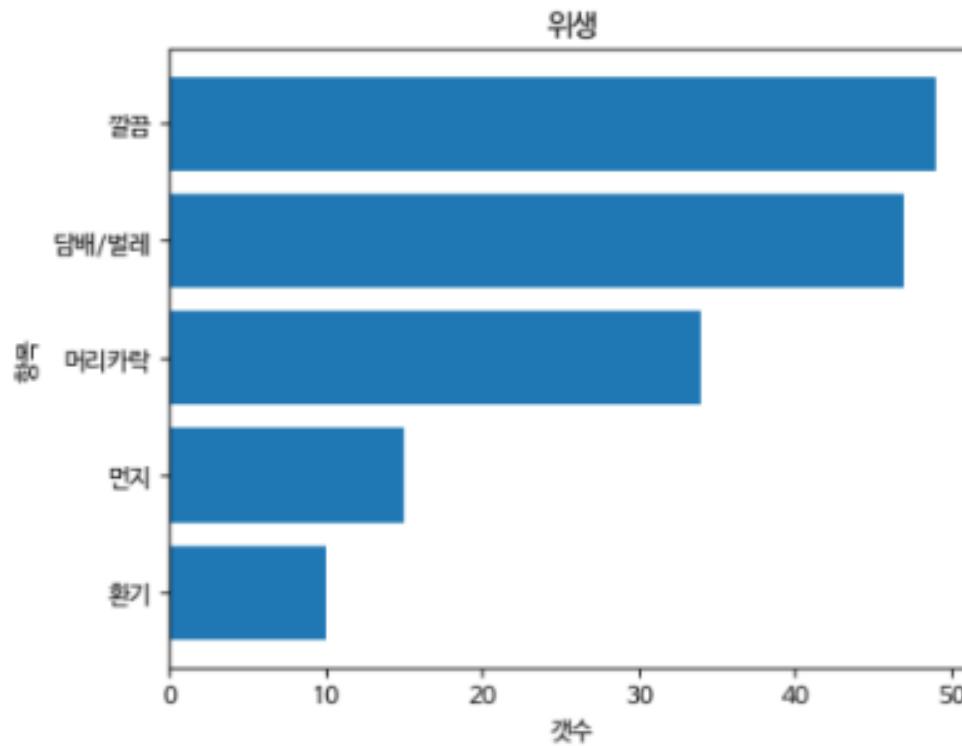
깔끔	145
청결	45
머리카락	37
담배	29
벌레	22
환기	18
먼지	15



- 사용한 시각화 라이브러리 : Matplotlib
- 간단한 시각화를 통해 이해를 돋기 위함

[라벨링한 키워드를 **분석과 함께 시각화!**]





키워드 : 위생

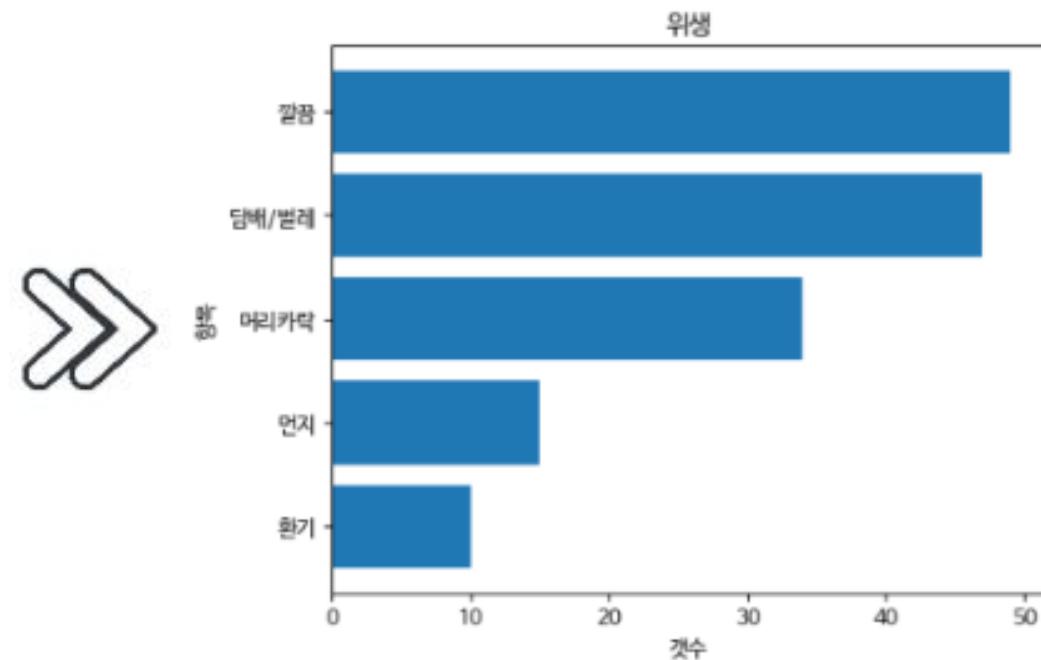
라벨링 단어 :

- 깔끔
- 담배/벌레
- 머리카락
- 먼지
- 환기

```
# 라벨링 단어들의 수 저장
count_clean1 = clean.count('담배') + clean.count('벌레')
count_clean2 = clean.count('먼지')
count_clean3 = clean.count('머리카락')
count_clean4 = clean.count('깔끔') + clean.count('정결')
count_clean5 = clean.count('환기')
```

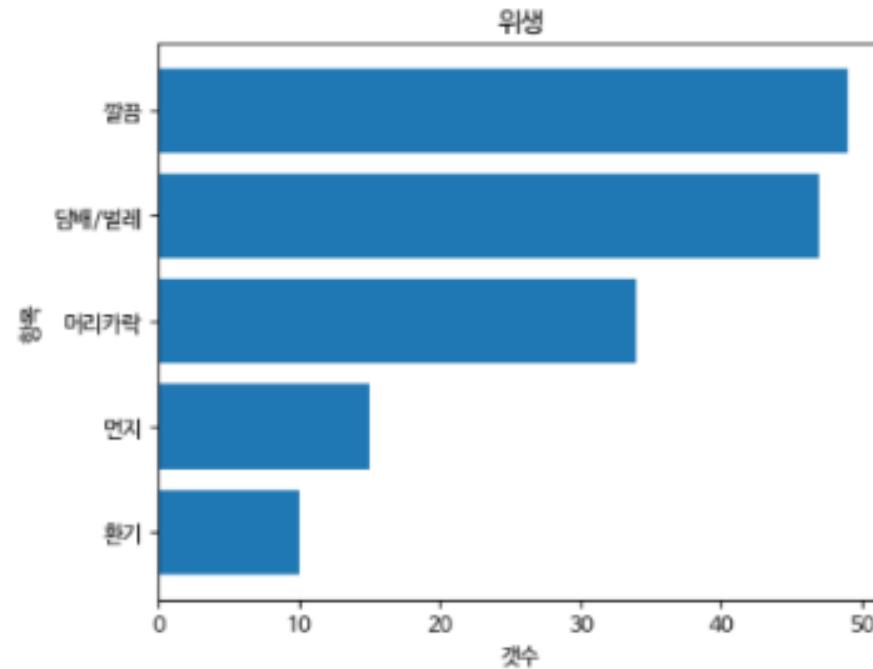
```
# 막대그래프 크기 순으로 정렬
categories = ['담배/벌레', '먼지', '머리카락', '깔끔', '환기']
counts = [count_clean1, count_clean2, count_clean3, count_clean4, count_clean5]
df_clean = pd.DataFrame({'index' : categories, 'val' : counts})
df_clean = df_clean.sort_values(by='val', ascending=True).reset_index(drop=True)
```

```
# 그래프 출력
ax[0][0].barh(df_clean.index, df_clean.val, color= 'tab:blue')
ax[0][0].set_xlabel('갯수')
ax[0][0].set_ylabel('항목')
ax[0][0].set_title('위생')
plt.show()
```





시각화한 그래프로 어떤 분석을?



'위생'이라는 키워드 안에서 리뷰 데이터 분석하기!!

1. 깔끔함을 언급하는 리뷰가 가장 많았음
2. 다음으로 담배, 벌레, 머리카락, 먼지 등 청결상태의 지적 리뷰들
3. 환기와 같은 사항들도 고려하자!

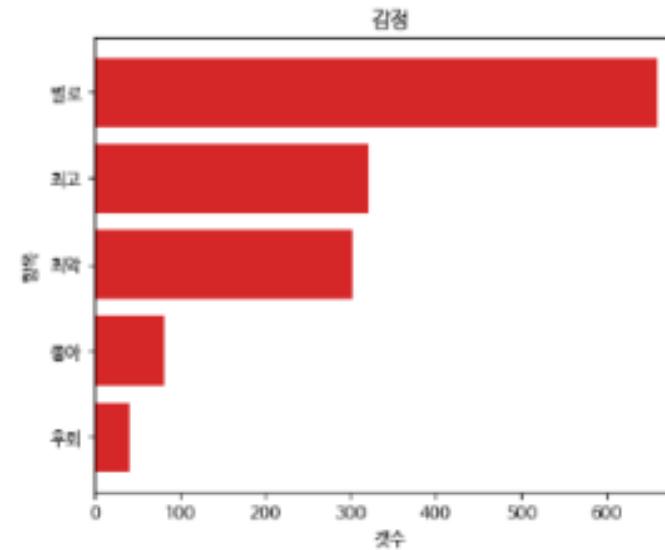
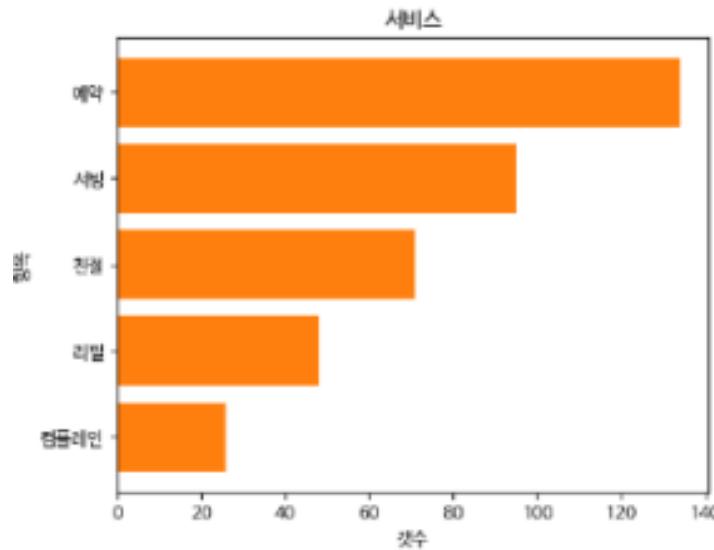


시각화한 그래프로 어떤 분석을?

- ✓ 매장의 깔끔함과 청결을 1순위로 생각
- ✓ 위생모나 앞치마를 착용 → 이물질을 차단
- ✓ 흡연 관련 대책을 강구할 필요성 인지함

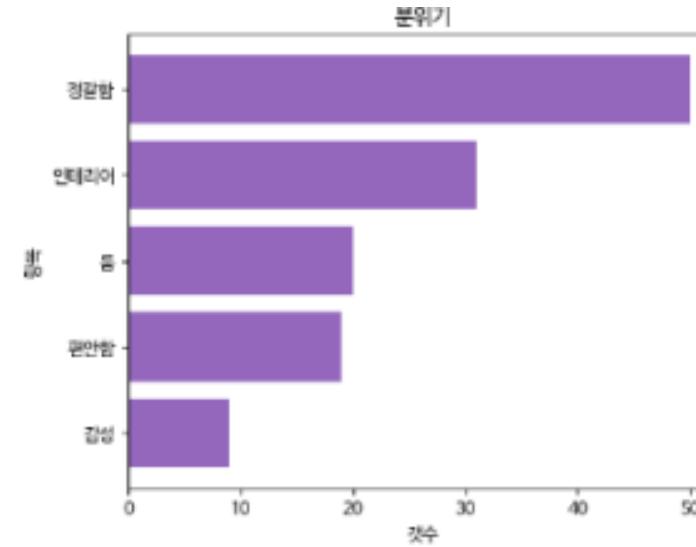
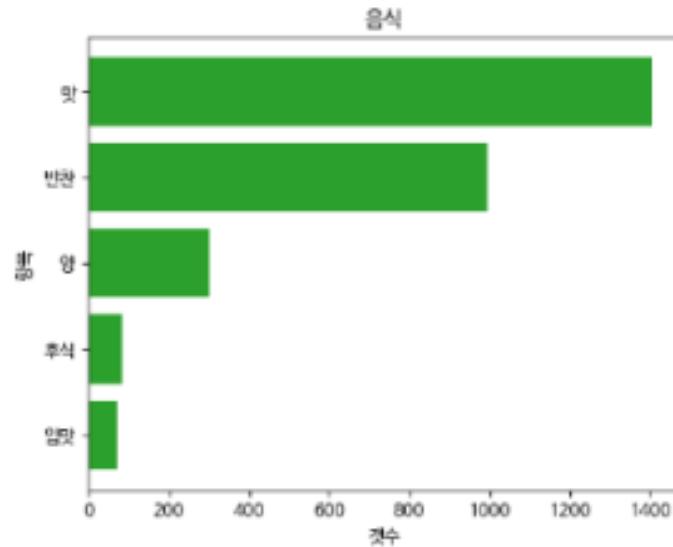


매출이 얹! 소리나는 가게!!!
전국구 맛집으로 입소문!!



- 예약 관련 서비스 최고 중요!
- 친절함은 기본 베이스~
- 커뮤니티에 잘 대처하는 매뉴얼

- 식당이 별로일때.. 난... 그 끔
... 리뷰를 쓴다..
- 최악보다는 최고가 되기위한 노력



- 식당의 가장 중요한 소양 → 맛
- 한식당의 특성 상 반찬 리뷰 ↑
- 넉넉한 인심 좋아요~

- 정갈한 분위기가 대세!
- 인스타감성 인테리어가 손님을 끌어 모은다고?

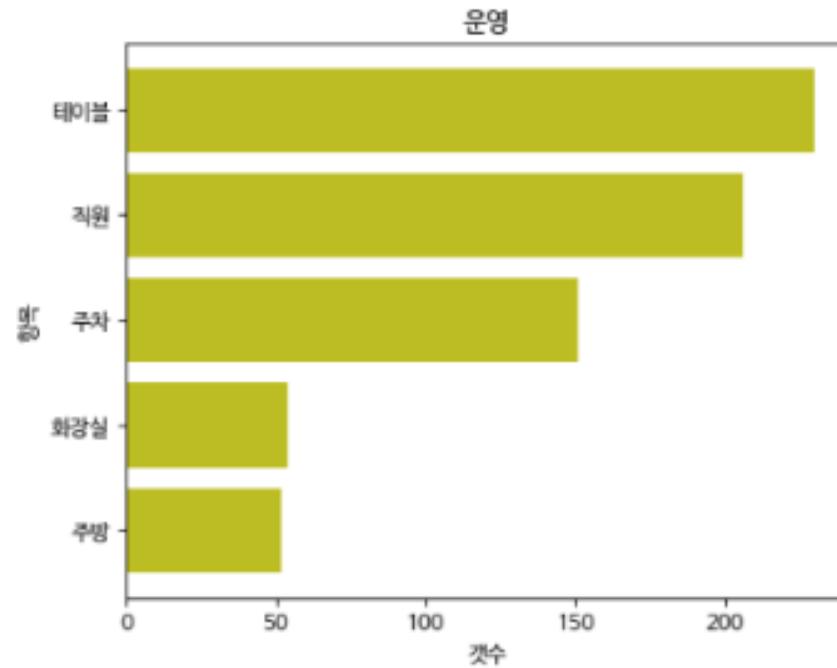


식당 운영에 있어 중요한 키워드는?





'운영', 운영을 잘 해야 오래 간다.



1. 테이블 관련 리뷰 수용 → 테이블간 거리, 테이블 청소상태 등
2. 직원관련 리뷰가 2순위 → 사람이 재산이다!
3. 주차공간 확보, 화장실 및 주방 관리도 중요!

03

기대 효과

잘하고 있어! Self-check List

- ☞ 식당 운영에서의 주요 키워드 기반으로 22개의 질문지 제작
- ☞ 총점 60점으로, 점수에 따른 등급 점검

50점 이상	 A 등급 - 고객 만족 보장하는 식당 고객들에게 훌륭한 경험을 제공하며 좋은 평가와 추천을 받는 인기식당입니다.
40점 이상	B 등급 - 고객 만족 보통 수준의 식당 대부분의 고객에게는 만족스러운 경험을 제공하지만, 일부 고객에게는 특별한 요소나 세부적인 사항이 부족할 수 있습니다. 부족한 키워드를 찾아 개선점을 찾을 필요가 있습니다.
40점 미만	C 등급 - 고객 만족이 낮은 식당 일반적으로 고객에게 실망과 불만을 제공할 가능성이 있습니다. 다양한 키워드 측면에서 기준에 미치 못하며, 이는 고객들이 추천하지 않을 수 있는 이유가 됩니다. 놓치고 있는 키워드를 확인하고 개선점을 찾아야 합니다.

한식당 체크 리스트				
Store Management Check Lists				
구분	점검사항			결과
		그렇다 (여기 25) 그지 30)	아니다 (0점)	
1	매장은 깨끗하게 유지되고 있나요?	<input checked="" type="checkbox"/>		
2	위생	주방 인원은 개인 위생에 신경 쓰고 있나요? ex) 하생모, 손不可避免, 알차비 등	<input checked="" type="checkbox"/>	
3		흡연 구역(부스) 관련 안내문이 있나요?	<input checked="" type="checkbox"/>	
4		설거지가 빠르고 깨끗이 이루어졌나요?		
5	서비스	예약 과정은 손님 입장에서 편하게 되어 있나요? 손님의 입장에서 예약을 시도해봤나요?		
6		전 직원들은 손님들께 친절하게 응대하고 있나요? 정기적으로 손님 응대에 대한 교육이 이뤄지고 있나요?		
7		팀 면담 회의 시 적정 명수로 운영되며 교류하게 되나요?		
8	품질	맛에 대한 고객의 평가는 좋은 상태로 유지되고 있나요?		
9		재료들은 모두 신선한 상태를 유지하고 있나요?		
10		주 요리에 알맞은 양은한분들이 준비되어 있나요?		
11		음식 재료에 부족함이 있나요?		
12		수자 시설이 완비 되어 있나요? 수자 모금은 합리적인가요?		
13		화장실 및 주방에 잘 관리되어 있나요?		
14		개계 내/외 환경 시설은 잘 정돈하고 있나요?		
15	환경	매장 내 테이블 간에 거리가 잘 차리지고 있나요?		
16		메이팅 방송 시, 관리할 수 있는 시스템이나 손님의 대기 공간이 준비되어 있나요?		
17		손님으로부터 접수된 불편 사항이 즉각 처리되고, 개인 사항이 유지되고 있나요?		
18		설비 환경에 맞게 실내 온도가 잘 유지되고 있나요?		
19	분위기	지정한 소년 및 인테리어 소품들이 잘 세팅 되었나요?		
20		개계의 정체성에 맞는 분위기가 잘 유지되고 있나요?		
21		단체 손님 방문 시 편안하게 석사할 수 있는 분위기 혹은 시설이 준비되어 있나요?		
총점 (60점)				

❶ 우리 가게 건강 진단표 (3분 소요)!

안녕하세요. 저희는 표시연으로 청연을 준비하고 계시거나 가게를 운영하시는 분들을 위해 간단한 체크리스트를 제작한 데이터 분석을 공부하고 있는 학생들입니다.

창업을 하고 계신 분들도, 또는 준비 중이신 분들께서 우리 가게 건강 진단을 할 수 있도록 각 질문의 문항들은 한국식당 리뷰 데이터를 분석 후 주요한 키워드를 도출하여 제작한 문항입니다.
마지막 총점 계산을 위해 점수를 기억해주세요!

* 제출해주시는 답변은 분석 목적 외 어떠한 품도로도 활용되지 않음을 안내 드립니다.

[위생] 내 가게의 청결도 진단!

* 해당하는 갯수만큼 중복 체크가 가능합니다.

문항 당 3점

매장은 깔끔하게 유지되고 있나요?

주방 인원은 개인 위생에 신경 쓰고 있나요? ex. 위생모, 손톱, 앞치마 등

흡연 구역(부스) 관련 안내문이 있나요?

설거지가 제대로 이루어 졌나요?

❷ 본격 우리 가게 건강 검진

[위생] 내 가게의 청결도 진단!

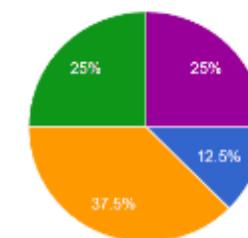
* 해당하는 갯수만큼 중복 체크가 가능합니다.



[위생] 위생 부문을 계산해주세요.

체크한 개수와 점수를 확인해보세요 :)

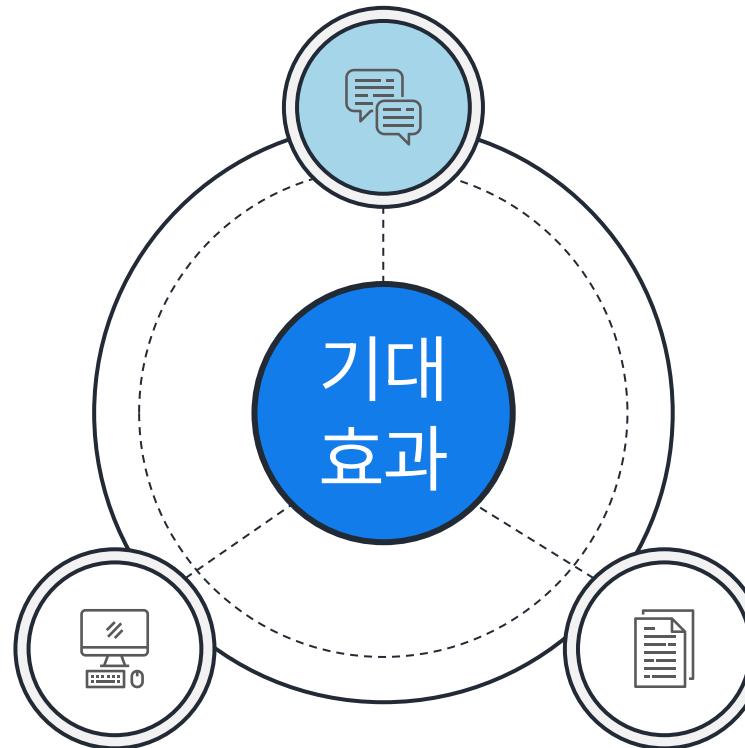
총점 8점



- 0점 → 총 0점
- 1점 → 총 3점
- 2점 → 총 6점
- 3점 → 총 9점
- 4점 → 총 12점

든든한 노후의 이정표

예비 창업자가 식당 창업을 준비하거나 운영 시 지표로 삼을 수 있는 키워드 제시



인사이트 제공

식당 운영의 모든 측면을 파악하고 효율성을 향상시킬 수 있으며 체크리스트를 작성하고 실제 운영과정에 적용하여 인사이트 획득 가능

개선점 발굴

한식당 창업에서 가장 필요한 키워드를 중심으로, 방문하는 고객이 기대하는 요소를 객관적으로 파악하여 궁극적으로 방문 만족도 증진과 건강한 자영업을 유지할 수 있음

04

회고

Keep

- 분석 목적을 반복적으로 상기하며 분석의 임팩트를 고민한 점
(ex. 키워드를 추출해서 얻을 수 있는 인사이트/기대효과는 무엇일까)
- 팀원의 코드가 풀리지 않을 때, 끈기 있게 고민하고 해결한 코드를 투명하게 공유한 점
- 예상과 다른 결과가 나오고 막혔을 때, 다른 방향성으로 빠르게 전환한 점
- 원활한 의사소통을 통해 빠른 주제 선정과 문제점 발생 시 수월한 대응이 가능했던 점
- 균형 있는 업무 분배와 어려움이 생겼을 때 각자가 할 수 있는 일을 찾아 역할에 충실한 점



Problem

- 선정한 데이터가 예상보다 데이터 분석에 적절하지 않았음
(ex. 긍정/부정이 나뉘어진 기준이 모호했던 점)
- 리뷰 데이터에서 가장 적절한 키워드를 선정하는 데 어려움
- 각종 형태소 분석기를 사용하여 시각화를 시도했으나, 두드러진 차별 요소가 없었고, 분석 없이도 예상했던 결과가 나옴
(ex. 식당에서 가장 중요한 것은 '친절', '맛')



Try

- 분석 데이터셋을 확정하기 전에 어떻게 활용할 건지 더 세부적인 논의 해보기
- 머신러닝 or 딥러닝 활용한 모델링 시도
- 다양한 시각화 라이브러리 사용



Q & A

감사합니다.