



# 리뷰데이터를 활용한 KoBert 기반 음식 추천 서비스

결정에 어려움을 겪고 있는 사람들을 위한 음식 결정 모델

2거 어때 | 홍세영 김소연 이준석 진세용

# index

## WHY

Overview

팀원 및 역할

## HOW

프로젝트 수행 절차

- Data

- Model

## RESULT

Web

Next Step

# Why

---

## | Overview : 기획 배경



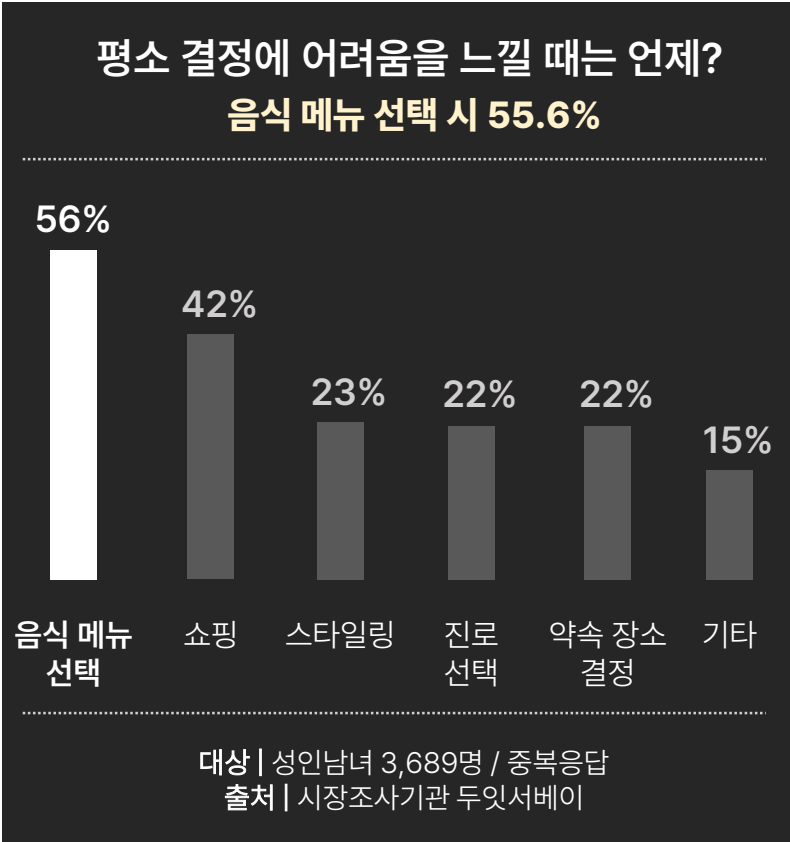
음식에 대한 수요와 종류가 다양해지면서 여러가지의 음식을 제공하는 맛집이 늘어났다.

음식을 선택할 수 있는 폭이 많아져 본인이 어떤 음식과 식당을 원하는지 고민해야 하는 문제점이 생겼다.

현대인의 최대 고민은 '음식 선택'이며, 우리는 매일 최소 1번 이상 '오늘 뭐 먹지?' 라는 고민을 한다.

**2거 어때 팀**은 현대인들이 가지고 있는 음식에 대한 고민을 줄여주고,  
식당을 선택하는데 있어 합리적인 의사결정에 도움을 주고자 **음식 추천 서비스** 프로젝트를 기획했다.

## | Overview : 결정에 어려움을 겪고 있는 사람들을 위해서



### 문제 인식

- 1 음식 메뉴 선택에 있어 많은 어려움**  
통계에 따르면 평소 음식 메뉴를 결정하는데 있어 사람들이 지나치게 많은 어려움을 겪고 있음
- 2 결정에 대한 부정적인 인터넷 신조어 발생**  
'결정 장애', '선택 불가 증후군', '햄릿 증후군'과 같은 인터넷 신조어 발생
- 3 지나치게 많은 선택지**
  - 현대인들은 음식을 선택하는데 있어 어려움을 겪고 있음
  - 지나치게 많은 선택지 때문에 뇌 과부하 발생

### 결론

음식 메뉴 선택에 어려움을 겪고있는 사람을 대상으로 합리적인 의사결정에 도움을 줄 수 있는 서비스의 필요성을 느낌

# Why

## | Overview : 서비스 간단 예상도

사용자의 자연어 Input



어제 술을 마셨더니  
매콤하고 칼칼한 음식이 먹고 싶어

서비스 Output



한식이 가장 적합한 음식입니다.

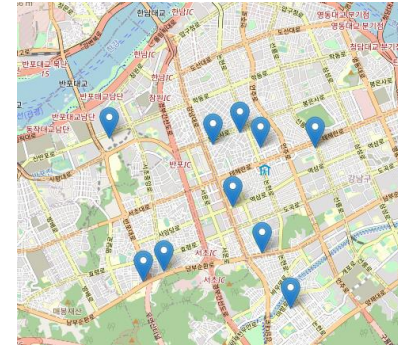
입력 문장과 일치하는  
음식 TOP3 입니다.

TOP1 한식  
TOP2 전골  
TOP3 분식

지도 API 기반 음식점 추천



근처의 한식 음식점 리스트입니다.



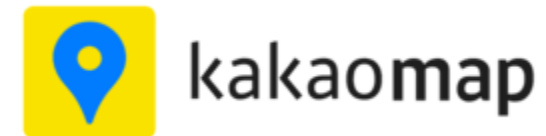
# Why

---

| Overview : 수행 도구 및 사용 언어



**Mecab**



# Why

---

## I 팀원 소개 및 역할 분담

공통 작업 | 데이터 수집, 데이터 전처리, 형태소 분석



홍세영 | 전체 총괄  
| 프로젝트 관리



김소연 | 웹 서비스 구현  
| PPT 제작



이준석 | API 연동  
| 웹 서비스 구현



진세용 | 모델 구축  
| 모델 평가

# part1. DATA



데이터 수집 | 데이터 전처리 | 데이터 탐색 & 분석 | 형태소 분석



| Data : 데이터 선정 과정 -어떤 리뷰데이터가 좋을까?

‘매콤’, ‘달달’, ‘담백한’과 같은 음식에 대한 맛 표현이 드러나있는 리뷰 데이터 필요



<네이버 마이플레이스>  
실 방문자 기반 평가 서비스



치즈폼칩초코스무디  
사진리뷰 53 · 1.23.월

정말 오랜만에 갔는데 여긴 정말 모든것이 그대로네요 역시 빠네가 제일 맛있어요

가성비가 좋아요



요밍미  
사진리뷰 25 · 2.2.목

런치세트로 피자랑 리조또 먹었는데 간이 센걸 좋아하는 제 입맛에는 리조또가 더 맛있었고 피자는 그저그랬어  
용 담에 온다면 리조또랑 뇨끼 먹어보고 싶네용! 청포도 에이드 맛있어요

인테리어가 멋져요 +3

플랫폼의 주제가 음식점에 한정되어 있지 않아  
원하는 정보를 가져오기 적합하지 않음



<배달 서비스>

gi\*\*님 2022년 12월 12일

★★★★★ | 맛 ★ 5 양 ★ 5 배달 ★ 5

올미트 오리지널 크러스트 R/1  
비오는 날인데도 배달 빨라서 좋았어요

al\*\*님 2022년 12월 23일

★★★★★ | 맛 ★ 5 양 ★ 5 배달 ★ 5

스노우볼 오리지널 크러스트 L/1,코카-콜라 제로 1.25L/1  
파파존스 짱짱!!!!

cs\*\*님 2022년 12월 18일

★★★★★ | 맛 ★ 5 양 ★ 5 배달 ★ 5

이탈리안 파파디아즈/1(추가 선택(파파스 파스타 ( 미트 )))칼릭 소스/1  
배달도 빠르고 맛있었습니다

대체적으로 배달에 관한 리뷰가 많았으며  
1~2줄의 짧은 데이터로 이루어져 있어  
적합하지 않음



<망고플레이트>  
리뷰 기반 맛집 평가 서비스



2023-01-11  
Hot & Crispy 그 자체..!

캐주얼 하고 힙한 분위기의 미국식 피자집. 여태 가본 미국식 피자집 중에서 가장 맛있게 먹고 나온  
곳이 아닐까 싶다. 가게 외관에 'Hot & Crispy', 'Oven Fresh', 'Large Slice'라고 쓰여져있는데 이  
곳의 피자가 놀랄케도 이 세가지 모두를 지지고 있었다.

얇은 도우, 크런치한 크러스트, 과하게 짜지 않은 토핑!

개인적으로 두꺼운 도우와 크러스트를 싫어하는 편이라 국내 프랜차이즈 피자집에서는 대부분 크러  
스트를 먹지 않고 남긴다. 이탈리아 화덕피자를 더 선호하는 이유이기도 하다.



6일 전  
식어도 맛있는 피자

매덕스 이름은 많이 들어봤지만 이상하게 지금까지 찾아가지 않은 곳인데요. 연말에 시음회 할 때 안  
주로 구입해보았습니다. 배달도 가능해서 시음회 장소에 배달로 주문해서 받았어요.

총 6종류의 피자를 주문했는데 다 맛있더라고요. 크기도... 엄청 컸어요. 도우가 얇아서 그래도 먹음  
만했지만 양이 꽤 많더라고요. 가서 먹은 게 아니고 배달로 주문해서 먹어서 다 식어 왔는데 식어서  
딱딱해지거나 질것하거나 흐물흐물하고 적당함(?) 식감을 유지하는 점이 맘에 들었습니다. 위에 올  
라간 토핑이나 소스도 맛있었구요.

음식 키워드 검색이 용이하며  
5줄 이상의 긴 리뷰와  
맛 표현이 포함된 리뷰들이 많았기에  
망고플레이트 리뷰 데이터를 사용하기로 함

## | Data : 망고플레이트 리뷰 데이터 크롤링

### 01 음식 카테고리 분류

#### 강남역 맛집 인기 검색순위

한식 | 분식 | 양식 | 세계음식 | 뷔페 | 디저트 | 카페 | 술집 | 치킨 | 브런치 | 이탈리아

#### 강남역 한식 맛집 인기 검색순위

한정식 | 해장국 | 해산물 | 칼국수 | 간장게장 | 냉면 | 전골 | 낙지 | 국수 |

<망고플레이트>의 음식 분류 기준을 통해 카테고리를 분류  
총 30개의 카테고리 생성

한정식 / 해장국 / 해산물 / 칼국수 / 간장게장 /  
냉면 / 전골 / 곰창 / 국수 / 떡볶이 /  
김밥 / 순대 / 샌드위치 / 버거 / 타코 /  
쌀국수 / 월남쌈 / 케이크 / 아이스크림 / 마카롱 /  
스시 / 라멘 / 생선회 / 돈가츠 / 베이커리 /  
통닭 / 파스타 / 피자 / 중식 / 마라탕

### 02 크롤링 코드 작성

```
# 검색어 입력 및 조회
search_name = '피자'
input_box = driver.find_element(By.CSS_SELECTOR, 'input.HomeSearchInput')
input_box.clear()
input_box.send_keys(search_name)
input_box.send_keys(Keys.ENTER)
time.sleep(1)

#리뷰들을 모을 리스트 생성
review_box = []

#페이지
for k in range(1, page_num + 1):
    page = f'body > main > article > div.column-wrapper > div > div > section > div.paging-container > p > a:nth'
    driver.find_element(By.CSS_SELECTOR, page).click()
    time.sleep(1)
    print(k, "페이지를 불러옵니다.")
    #행
    for j in range(1, rows + 1):
        #j번으로 페이지넘기기
        selc = f'body > main > article > div.column-wrapper > div > div > section > div.search-list-restaurants'
        driver.find_element(By.CSS_SELECTOR, selc).click()
        time.sleep(2)
        print(j, "번째 행 크롤링을 시작합니다.")
        #열
        for i in range(1, 3):
            #j번째 행 i번째 크롤링을 시작
            selector = f'body > main > article > div.column-wrapper > div > div > section > div.search-list-res'
            driver.find_element(By.CSS_SELECTOR, selector).click()
            time.sleep(2)
            print(j, "번째 행", i, "번째 크롤링을 시작합니다.")
            #리뷰 수 파악
            review_num = driver.find_element(By.CSS_SELECTOR, 'body > main > article > div.column-wrapper > div'
            review_num = int(review_num)
            #마지막 리뷰가 나올때까지 '더보기' 클릭 #스크롤 내리기
            last_height = driver.execute_script("return document.body.scrollHeight") # 현재 높이를 알려줌
            while True:
                # 스크롤다운
```

Selenium을 이용하여 크롤링  
앞서 생성한 카테고리별로 최대 10페이지의 리뷰 수집

## ■ Data : 망고플레이트 리뷰 데이터 크롤링

- Raw review data\_dataframe

Raw review data\_histogram

## | Data : 데이터 전처리

- 기호, 특수문자, 이모티콘, 잘못된 띄어쓰기, 공백, 영어, 숫자 제거
- Null값 제거
- 중복 데이터 값 제거
- 숫자로 되어있던 label(카테고리)를 한글로 변환



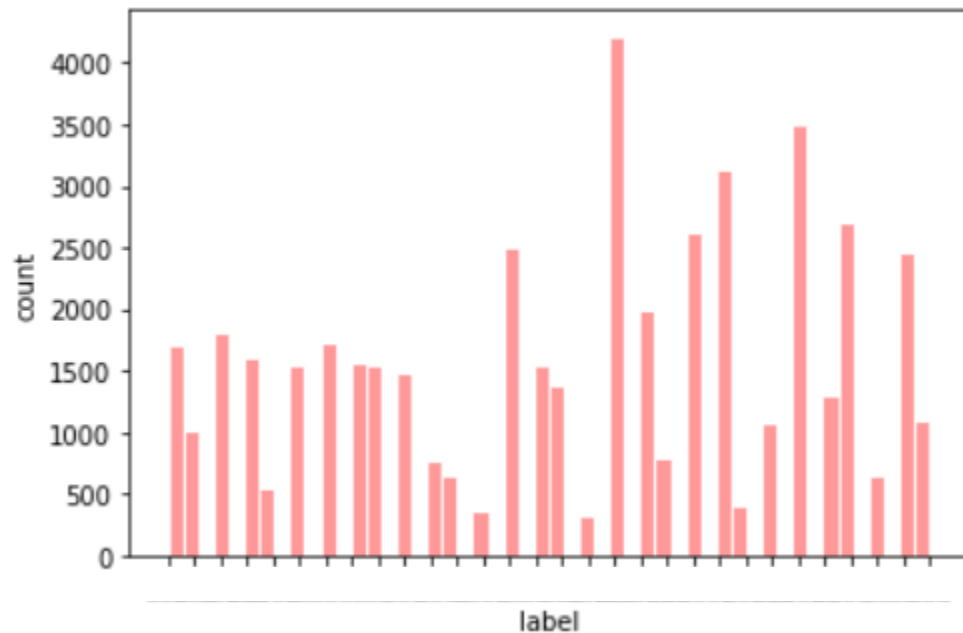
- 총 48,018 개의 리뷰 데이터 확보
- 앞서 분류한 30개의 카테고리를 각각의 데이터에 라벨링
- 카테고리 각각의 리뷰 수 분포 확인
  - Max review: 18번 카테고리 - 케이크(4,219)
  - Min review: 23번 카테고리 - 생선회 (321)

Preprocessed review data\_dataframe

	reviews	label_kor
index		
0	힙지로에서는 조금 떨어진 하지만 금요일 저녁에 직장인들이 술 먹기엔 좋은 곳 날씨도...	한정식
1	힐링이 필요할 때 가는 곳 힐링이 필요할 때 가는 곳	한정식
2	희락갈치는 남대문 갈치골목에 위치한 인기있는 항상 사람이 가득한거 같은 식당입니다 ...	한정식
3	홍선대원군의 별장이었던 석파정 근처의 석파랑은 그 스토리와 외관 등 여러가지 이유로...	한정식
4	흑돼지덮밥 존맛이당 흑돼지덮밥 존맛이당	한정식
...	...	...
48013	사골마라탕 대학로에서는 마라탕으로 여기를 이기기 쉽지 않다 정말 한글보다 중국어가 ...	마라탕
48014	마라탕 고기추가 적당히 대중적이고 무난한 마라탕 대학로에서 마라탕 맛집 찾는거 정말...	마라탕
48015	마귀같은 음식점 음식에 뭐를 넣었는지 자꾸 생각나서 들어가기 전 주기도문을 외우게 ...	마라탕
48016	한창 마라상귀에 꽃혀 있을 때 여기만 갔었어요 지하에 있지만 매장도 깔끔한 편이고 ...	마라탕
48017	중경상림이나 화양연화에서 나올 법한 홍콩 분위기 힙한 퓨전 중식 레스토랑 대체로 맥...	마라탕

48018 rows x 2 columns

Preprocessed review data\_histogram



## | Data : 카테고리 별 키워드 추출 -형태소 분석기 선택 과정

### 01

#### 정규화 작업 진행

- 키워드 추출을 위한 정규화 작업 필요
- 정규화 작업이란 같은 뜻이지만 표현 방법이 다른 단어들을 통합시켜 같은 단어로 만드는 것

### 02

#### 표제어 추출 필요

- 단어들로 부터 표제어를 찾아가는 과정
- 표제어 추출이란 단어의 뿌리 단어를 찾아 단어의 근본적 의미를 찾아내는 것

#### 표제어 추출 선정 이유

- 어간 추출은 단어의 형태가 바뀌기 때문에 품사 정보가 변형되거나 정의되지 않을 수 있음
- 표제어 추출은 어간 추출과 달리 문맥을 고려하여 해당 단어의 품사 정보를 보존 가능

### 03

#### 형태소 분석기 Okt와 MeCab 비교

- Khma와 Komoran과 같은 어간 추출에 특화된 다른 형태소 분석기와는 달리 Okt와 MeCab은 표제어 추출에 용이
- 그 외 사용자 사전 기능, 분석 처리 속도, 키워드를 추출하는데 적합한 분석 방식을 가지고 있는 두 형태소 분석기를 비교하기로 결정

| Data : 카테고리 별 키워드 추출 -형태소 분석

형태소 분석기 MeCab 채택

- Okt와 Mecab을 사용해보고 장단점을 비교한 결과 Okt의 경우에 명사를 하나로 묶어서 표현하지만, Mecab의 경우 명사를 조금 더 세분화 해서 표현
- 처리속도가 월등히 빠르며 현업에서 사용하는 비율이 높음

'가성비',,,NNP,\*F,가성비,\*,\*,\*,\*\n',  
'공기밥',,,NNP,\*T,공기밥,\*,\*,\*,\*\n',  
'깍두기',,,NNP,\*F,깍두기,\*,\*,\*,\*\n',  
'꿔바로우',,,NNP,\*F,꿔바로우,\*,\*,\*,\*\n',  
'낙곱새',,,NNP,\*T,낙곱새,\*,\*,\*,\*\n',  
'남자 친구',,,NNP,\*F,남자 친구,\*,\*,\*,\*\n',  
'닭도리탕',,,NNP,\*T,닭도리탕,\*,\*,\*,\*\n',  
'델리만쥬',,,NNP,\*F,델리만쥬,\*,\*,\*,\*\n',  
'도토리',,,NNP,\*F,도토리,\*,\*,\*,\*\n',  
'돈까스',,,NNP,\*F,돈까스,\*,\*,\*,\*\n',  
'도넛',,,NNP,\*F,도넛,\*,\*,\*,\*\n',  
'족배기',,,NNP,\*F,족배기,\*,\*,\*,\*\n',  
'라면',,,NNP,\*T,라면,\*,\*,\*,\*\n',  
'리조또',,,NNP,\*F,리조또,\*,\*,\*,\*\n',  
'마카롱',,,NNP,\*T,마카롱,\*,\*,\*,\*\n',  
'맛 집',,,NNP,\*T,맛 집,\*,\*,\*,\*\n',  
'맥앤치즈',,,NNP,\*F,맥앤치즈,\*,\*,\*,\*\n',  
'모짜렐라',,,NNP,\*F,모짜렐라,\*,\*,\*,\*\n',  
'몬테크리스토',,,NNP,\*F,몬테크리스토,\*,\*,\*,\*\n',  
'말피유니베',,,NNP,\*T,말피유니베,\*,\*,\*,\*\n',  
'반세오',,,NNP,\*F,반세오,\*,\*,\*,\*\n',  
'번데기',,,NNP,\*F,번데기,\*,\*,\*,\*\n',  
'부대 찌개',,,NNP,\*T,부대 찌개,\*,\*,\*,\*\n',  
'부대찌개',,,NNP,\*T,부대찌개,\*,\*,\*,\*\n',  
'부리또',,,NNP,\*F,부리또,\*,\*,\*,\*\n',  
'샌드위치',,,NNP,\*F,샌드위치,\*,\*,\*,\*\n',  
'세꼬시',,,NNP,\*F,세꼬시,\*,\*,\*,\*\n',  
'소보루',,,NNP,\*F,소보루,\*,\*,\*,\*\n',  
'수제비',,,NNP,\*F,수제비,\*,\*,\*,\*\n',  
'쉬림프',,,NNP,\*F,쉬림프,\*,\*,\*,\*\n',  
'슴슴한',,,NNP,\*T,슴슴한,\*,\*,\*,\*\n',  
'아이스크림',,,NNP,\*T,아이스크림,\*,\*,\*,\*\n',  
'아인슈페너',,,NNP,\*F,아인슈페너,\*,\*,\*,\*\n',  
'알버터',,,NNP,\*F,알버터,\*,\*,\*,\*\n',

0	가
1	가게
2	가격
3	가격대
4	가까우
...	...
2042	히츠
2043	힘
2044	힘내
2045	힘드
2046	힘들

2047 rows x 1 columns

| 키워드 추출을 위한 준비 과정

1. 사용자 사전 등록

Mecab이 이해하지 못하는 신조어를 사용자 사전에 등록

2. 불용어 제거 처리

키워드 추출을 위해 음식과 맛에 대한 음절이 아닌 경우 모두 불용어 처리

# How

---

| Data : 형태소 분석을 통한 카테고리 별 키워드 추출

카테고리 별 형태소가 명사, 동사, 형용사, 어근인 단어 추출



카테고리 별 빈도가 높은 키워드를 추출하기 위해 명사와 동사/형용사/어근 으로 나누어 2번 분석

형태소 중 명사의 빈도가 월등히 높아 동사/형용사/어근인 단어를 추출하기 위해 명사를 따로 나누어 분석 진행



태그 별 빈도가 높은 키워드 20개씩 선정하여 각 카테고리 별 총 40개의 키워드 추출



총 48,018 개의 확보된 리뷰 데이터에서 키워드가 포함된 문장만 추출

# How

## | Data : 형태소 분석을 통한 카테고리 별 키워드 추출

Keyword

	keyword	counts	tag	label
0	계장	728	명사	한정식
1	만두	695	명사	한정식
2	반찬	651	명사	한정식
3	김치	632	명사	한정식
4	간장	533	명사	한정식
...	...	...	...	...
1195	칼칼	28	동형어	마라탕
1196	딱딱	28	동형어	마라탕
1197	촉촉	27	동형어	마라탕
1198	밍밍	27	동형어	마라탕
1199	얼큰	26	동형어	마라탕

1200 rows × 4 columns

Keyword 포함 문장 추출

		reviews	label_kor
index			
0	힙지로에서는 조금 떨어진 하지만 금요일 저녁에 직장인들이 술 먹기엔 좋은 곳 날씨도...		한정식
2	희락갈치는 남대문 갈치골목에 위치한 인기있는 항상 사람이 가득한거 같은 식당입니다 ...		한정식
3	흥선대원군의 별장이었던 석파정 근처의 석파랑은 그 스토리와 외관 등 여러가지 이유로...		한정식
4	흑돼지덮밥 존맛이당 흑돼지덮밥 존맛이당		한정식
5	흑돼지 편백찜 원 인분 팽이버섯 추가 깻잎 명이나물에 너무 맛있게 잘 싸먹었어요 와...		한정식
...		...	...
48013	사골마라탕 대학로에서는 마라탕으로 여기를 이기기 쉽지 않다 정말 한글보다 중국어가 ...		마라탕
48014	마라탕 고기추가 적당히 대중적이고 무난한 마라탕 대학로에서 마라탕 맛집 찾는거 정말...		마라탕
48015	마귀같은 음식점 음식에 뭐를 넣었는지 자꾸 생각나서 들어가기 전 주기도문을 외우게 ...		마라탕
48016	한창 마라상귀에 꽃혀 있을 때 여기만 갔었어요 지하에 있지만 매장도 깔끔한 편이고 ...		마라탕
48017	중경상림이나 화양연화에서 나올 법한 홍콩 분위기 힙한 퓨전 중식 레스토랑 대체로 맥...		마라탕

44977 rows × 2 columns

키워드 추출

전처리 된 데이터에서 키워드가 포함된 문장만 추출했을 때 데이터의 수의 변화가 극단적으로 크게 보이지 않았음  
48,018 개 -> 44,977 개



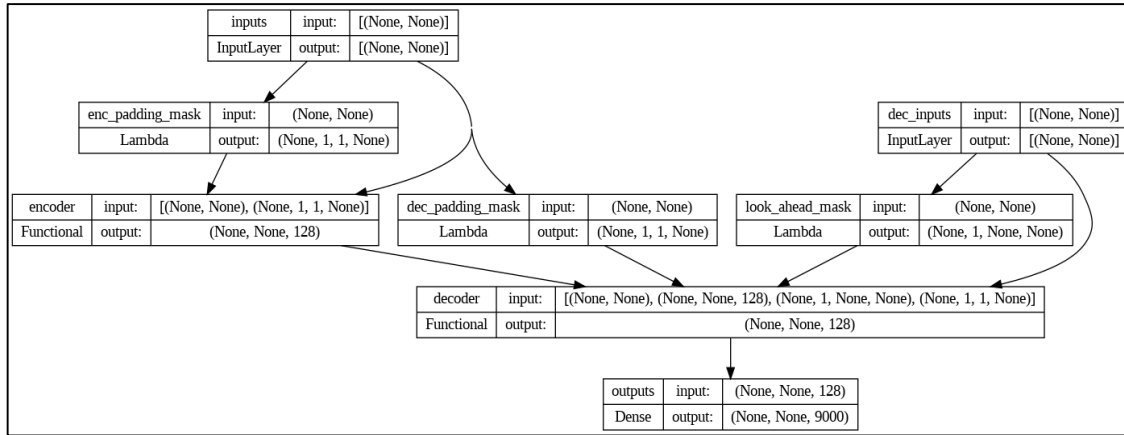
# part2. MODEL

Transformer | KoBert | 모델 평가

# How

## Model : Transformer

### 선정 이유



### 결과값

```
[ ] EPOCHS = 10
model.fit(dataset, epochs=EPOCHS)

Epoch 1/10
358/358 [=====] - 30s 60ms/step - loss: 0.2713 - accuracy: 0.0235
Epoch 2/10
358/358 [=====] - 20s 57ms/step - loss: 0.0719 - accuracy: 0.0300
Epoch 3/10
358/358 [=====] - 21s 58ms/step - loss: 0.0382 - accuracy: 0.0401
Epoch 4/10
358/358 [=====] - 21s 58ms/step - loss: 0.0272 - accuracy: 0.0431
Epoch 5/10
358/358 [=====] - 21s 59ms/step - loss: 0.0211 - accuracy: 0.0449
Epoch 6/10
358/358 [=====] - 21s 58ms/step - loss: 0.0164 - accuracy: 0.0462
Epoch 7/10
358/358 [=====] - 21s 59ms/step - loss: 0.0130 - accuracy: 0.0473
Epoch 8/10
358/358 [=====] - 21s 59ms/step - loss: 0.0102 - accuracy: 0.0481
Epoch 9/10
358/358 [=====] - 21s 59ms/step - loss: 0.0087 - accuracy: 0.0485
Epoch 10/10
358/358 [=====] - 21s 58ms/step - loss: 0.0075 - accuracy: 0.0490
<keras.callbacks.History at 0x7f3feb630fd0>
```

```
[ ] # 테스트1
output = predict("바삭바삭")

Input: 바삭바삭
Output: 종식

[ ] # 테스트2
output = predict("반찬이 있고 불고기같은게 먹고싶어")

Input: 반찬이 있고 불고기같은게 먹고싶어
Output: 간장게장

[ ] # 테스트3
output = predict("달콤하고 차가운게 끌려")

Input: 달콤하고 차가운게 끌려
Output: 라면
```

- 초기에는 Transformer와 Bert를 두고 고민했지만, 큰 기축이 되었던 토대 모델이었기에 Transformer를 먼저 사용
- 트랜스포머 모델은 문장 속 단어와 같은 순차 데이터 내의 관계를 추적해 맥락과 의미를 학습하는 신경망이며, 어텐션 & 셀프-어텐션 등을 이용하여 서로 떨어져 있는 부분의 관계 또한 미묘하게 감지해 내는 특징
- 입력 시퀀스가 길어져 출력의 정확도가 떨어지는 것을 보정해주는 어텐션 기법을 적용한 Transformer모델을 통해서 입력 시퀀스가 긴 리뷰데이터의 출력 시퀀스 정확도 하락을 보정해 줄 것이라고 판단

- '전처리 -> 형태소 분석-> 모델 학습'의 순서로 처음 학습을 시켰을 때, 약 3%의 정확도 도출  
➤ 전처리 & 형태소 분석의 문제로 가정
- 전처리를 여러 방법으로 시도해본 모델 학습, 형태소 분석의 카테고리 별 키워드 파라미터 값을 다르게 설정한 모델 학습의 결과 정확도는 약 5%를 넘지 못하는 결과 도출
- 테스트 과정에서 입력 문장에 대한 Output확인 결과  
➤ 정확도는 낮았지만 문장 Test에서 생각보다 문맥에 맞는 Output을 도출
- 위 결과에 따라 모델과 데이터의 문제라고 판단

# How

## | Model : Transformer

```
[ ] EPOCHS = 10
model.fit(dataset, epochs=EPOCHS)

Epoch 1/10
358/358 [=====] - 30s 60ms/step - loss: 0.2713 - accuracy: 0.0235
Epoch 2/10
358/358 [=====] - 20s 57ms/step - loss: 0.0719 - accuracy: 0.0300
Epoch 3/10
358/358 [=====] - 21s 58ms/step - loss: 0.0382 - accuracy: 0.0401
Epoch 4/10
358/358 [=====] - 21s 58ms/step - loss: 0.0272 - accuracy: 0.0431
Epoch 5/10
358/358 [=====] - 21s 59ms/step - loss: 0.0211 - accuracy: 0.0449
Epoch 6/10
358/358 [=====] - 21s 58ms/step - loss: 0.0164 - accuracy: 0.0462
Epoch 7/10
358/358 [=====] - 21s 59ms/step - loss: 0.0130 - accuracy: 0.0473
Epoch 8/10
358/358 [=====] - 21s 59ms/step - loss: 0.0102 - accuracy: 0.0481
Epoch 9/10
358/358 [=====] - 21s 59ms/step - loss: 0.0087 - accuracy: 0.0485
Epoch 10/10
358/358 [=====] - 21s 58ms/step - loss: 0.0075 - accuracy: 0.0490
<keras.callbacks.History at 0x7f3feb630fd0>
```

- 입력 시퀀스가 길어져 출력의 정확도가 떨어지는 것을 보정해주는 어텐션 기법을 적용한 Transformer 모델을 통해서 입력 시퀀스가 긴 리뷰데이터의 출력 시퀀스 정확도 하락을 보정해 줄 것이라고 판단

```
[ ] # 테스트1
output = predict("바삭바삭")

Input: 바삭바삭
Output: 중식

[ ] # 테스트2
output = predict("반찬이 있고 불고기같은게 먹고싶어")

Input: 반찬이 있고 불고기같은게 먹고싶어
Output: 간장게장

[ ] # 테스트3
output = predict("달콤하고 차가운게 끌려")

Input: 달콤하고 차가운게 끌려
Output: 라멘
```

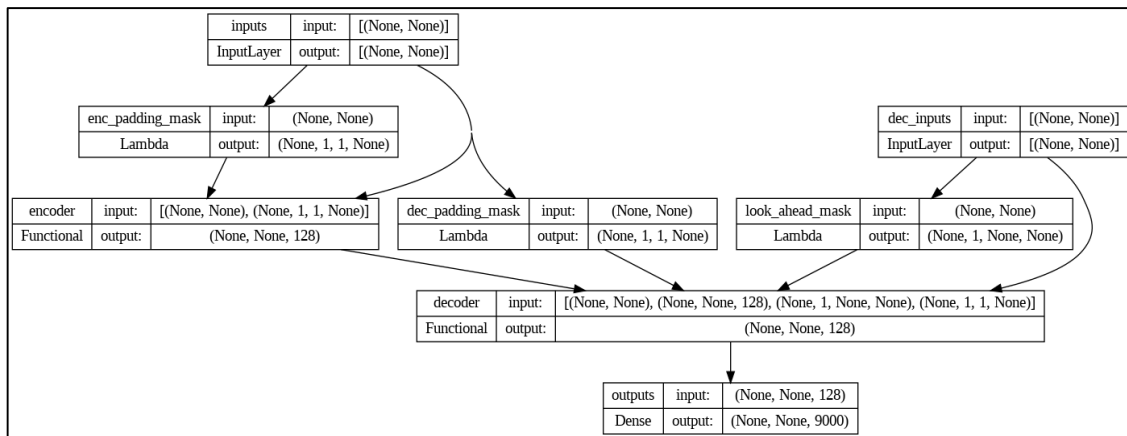
결과 도출

- 테스트 과정에서 입력 문장에 대한 Output 확인 결과  
➤ 정확도는 낮았지만 문장 Test에서 생각보다 문맥에 맞는 Output을 도출
- 위 결과에 따라 모델과 데이터의 문제라고 판단

# How

## Model : Transformer

### 선정 이유



### 결과값

```
[ ] EPOCHS = 10
model.fit(dataset, epochs=EPOCHS)

Epoch 1/10
358/358 [=====] - 30s 60ms/step - loss: 0.2713 - accuracy: 0.0235
Epoch 2/10
358/358 [=====] - 20s 57ms/step - loss: 0.0719 - accuracy: 0.0300
Epoch 3/10
358/358 [=====] - 21s 58ms/step - loss: 0.0382 - accuracy: 0.0401
Epoch 4/10
358/358 [=====] - 21s 58ms/step - loss: 0.0272 - accuracy: 0.0431
Epoch 5/10
358/358 [=====] - 21s 59ms/step - loss: 0.0211 - accuracy: 0.0449
Epoch 6/10
358/358 [=====] - 21s 58ms/step - loss: 0.0164 - accuracy: 0.0462
Epoch 7/10
358/358 [=====] - 21s 59ms/step - loss: 0.0130 - accuracy: 0.0473
Epoch 8/10
358/358 [=====] - 21s 59ms/step - loss: 0.0102 - accuracy: 0.0481
Epoch 9/10
358/358 [=====] - 21s 59ms/step - loss: 0.0087 - accuracy: 0.0485
Epoch 10/10
358/358 [=====] - 21s 58ms/step - loss: 0.0075 - accuracy: 0.0490
<keras.callbacks.History at 0x7f3feb630fd0>
```

```
[ ] # 테스트1
output = predict("바삭바삭")

Input: 바삭바삭
Output: 종식

[ ] # 테스트2
output = predict("반찬이 있고 불고기같은게 먹고싶어")

Input: 반찬이 있고 불고기같은게 먹고싶어
Output: 간장게장

[ ] # 테스트3
output = predict("달콤하고 차가운게 끌려")

Input: 달콤하고 차가운게 끌려
Output: 라면
```

- 초기에는 Transformer와 Bert를 두고 고민했지만, 큰 기축이 되었던 토대 모델이었기에 Transformer를 먼저 사용
- 트랜스포머 모델은 문장 속 단어와 같은 순차 데이터 내의 관계를 추적해 맥락과 의미를 학습하는 신경망이며, 어텐션 & 셀프-어텐션 등을 이용하여 서로 떨어져 있는 부분의 관계 또한 미묘하게 감지해 내는 특징
- 입력 시퀀스가 길어져 출력의 정확도가 떨어지는 것을 보정해주는 어텐션 기법을 적용한 Transformer모델을 통해서 입력 시퀀스가 긴 리뷰데이터의 출력 시퀀스 정확도 하락을 보정해 줄 것이라고 판단

- '전처리 -> 형태소 분석-> 모델 학습'의 순서로 처음 학습을 시켰을 때, 약 3%의 정확도 도출  
➤ 전처리 & 형태소 분석의 문제로 가정
- 전처리를 여러 방법으로 시도해본 모델 학습, 형태소 분석의 카테고리 별 키워드 파라미터 값을 다르게 설정한 모델 학습의 결과 정확도는 약 5%를 넘지 못하는 결과 도출
- 테스트 과정에서 입력 문장에 대한 Output확인 결과  
➤ 정확도는 낮았지만 문장 Test에서 생각보다 문맥에 맞는 Output을 도출
- 위 결과에 따라 모델과 데이터의 문제라고 판단

## | Model : Transformer 모델의 낮은 Accuracy 원인 파악

1

### 데이터 정제의 문제

초기 데이터 정제는 중복 값 제거, 공백 제거, 특수 기호 제거로 진행했던 초기 데이터에서 추가적으로 영문 제거와 자음&모음 제거

» **정확도 개선 실패**

2

### 형태소 분석 과정의 문제

형태소 분석 과정의 문제인지 파악하기 위해 형태소 분석을 한 데이터와 형태소 분석을 하지 않은 데이터로 정확도 측정

» **정확도 개선 실패**  
더 이상 데이터 정제의 문제는 아니라고 판단

3

### 모델 파라미터 값의 문제

모델을 구성하는 파라미터 값들에 차이를 두어 실험  
ex. max\_len, batch\_size, epochs, learning\_rate

» **정확도 개선 실패**  
정확도가 미미하게 향상되었지만 유의미한 수준은 아닌 것으로 판단

4

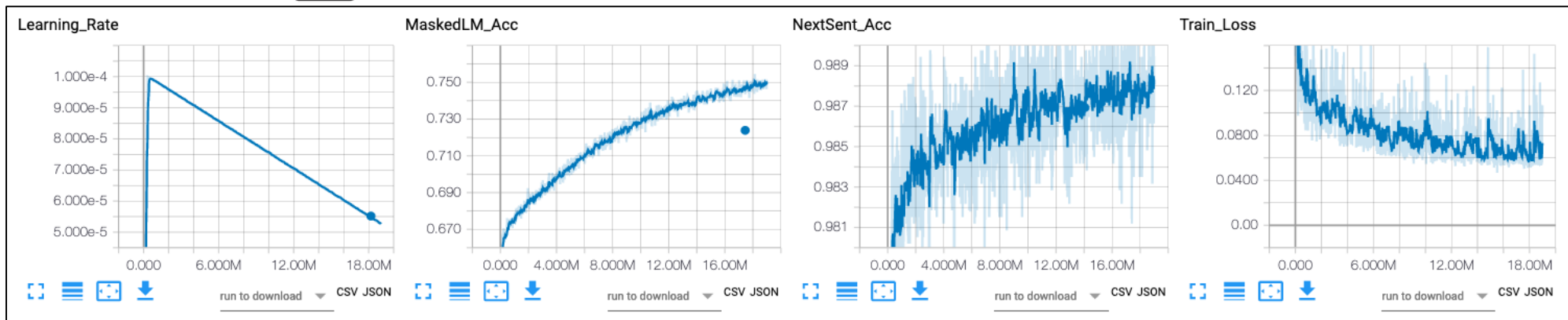
### 모델의 문제

- 30개의 카테고리 -> 17개 카테고리로 축소
- 다중 분류 모델인 Bert 계열의 학습 모델을 탐색하여  
    **최종적으로 KoBERT 모델을 선택**
- 새로운 학습 모델에 넣을 데이터를 다시 정제
- 카테고리 별 데이터 평준화 등의 데이터를 전처리

## | Model : 새로운 모델 탐색 -KoBERT

SKTBrain/**KoBERT**

Korean BERT pre-trained cased (KoBERT)



- KoBERT는 기존 BERT의 한국어 성능 한계를 극복하기 위해 개발
- 수백만 개의 한국어 문장으로 이루어진 대규모 말뭉치를 학습하여, 한국어의 불규칙한 언어 변화의 특성을 반영하기 위해 데이터 기반 토큰화 기법 적용 (기존 대비 27%의 토큰만으로 2.6% 이상의 성능 향상)
- 대량의 데이터를 빠른 시간에 학습하기 위해 링 리듀스 기반 분산 학습 기술을 사용하여, 빠르게 학습할 수 있는 점이 특징

문맥에 의존하는 특징적인 표현의 전이 학습을 실시한다는 KoBERT의 특징이 키워드 중심으로 카테고리를 표현하는 '2거 어때'팀의 프로젝트 방향성과 알맞음

## Model : KoBERT 모델 구축

### 모델 구축

```
[ ] class BERTClassifier(nn.Module):
    def __init__(self,
                  bert,
                  hidden_size = 768,
                  num_classes=17,
                  dr_rate=None,
                  params=None):
        super(BERTClassifier, self).__init__()
        self.bert = bert
        self.dr_rate = dr_rate

        self.classifier = nn.Linear(hidden_size , num_classes)
        if dr_rate:
            self.dropout = nn.Dropout(p=dr_rate)

    def gen_attention_mask(self, token_ids, valid_length):
        attention_mask = torch.zeros_like(token_ids)
        for i, v in enumerate(valid_length):
            attention_mask[i][:v] = 1
        return attention_mask.float()

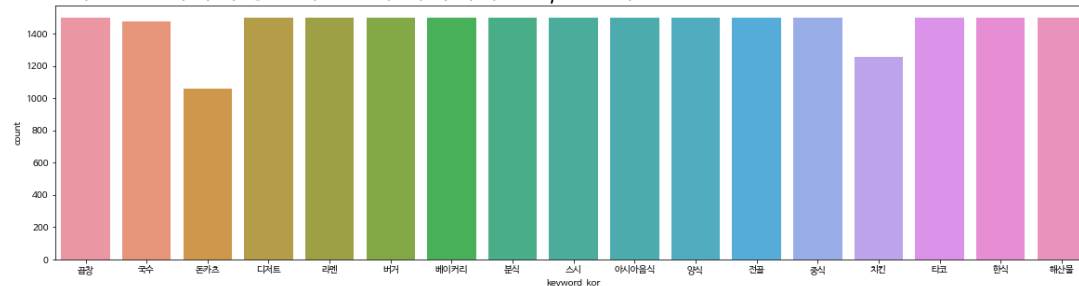
    def forward(self, token_ids, valid_length, segment_ids):
        attention_mask = self.gen_attention_mask(token_ids, valid_length)

        _, pooler = self.bert(input_ids = token_ids, token_type_ids = segment_ids)

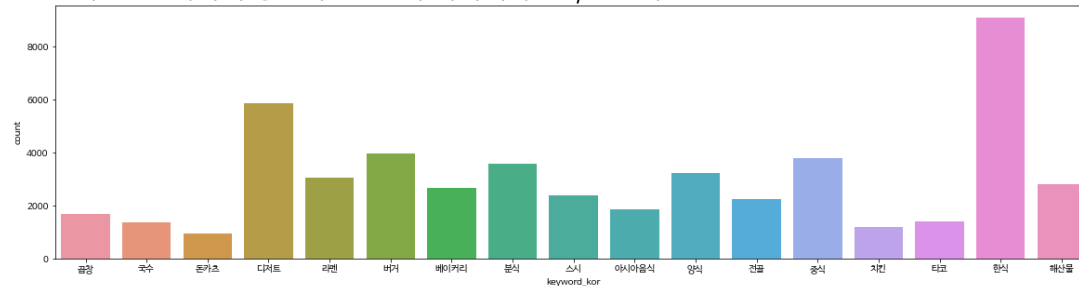
        if self.dr_rate:
            out = self.dropout(pooler)
        return self.classifier(out) #batchsize * num_classes
```

### 데이터 정제

1차 test\_데이터 평준화 : 전체 데이터 수 24,785개



2차 test\_데이터 평준화 X : 전체 데이터 수 51,034개



### 모델 학습

- 1차 시도로 데이터 평준화를 위해 1,500개씩 데이터 정제한 학습 결과 | **Train\_Acc = 0.93 / Test\_Acc = 0.70**
- Test\_data의 성능을 높이기 위해 파라미터 값 / 데이터 값을 수정하여 재학습 => 성능 개선 한계점 도달
- 2차 시도로 데이터 평준화를 하지 않은 학습 결과 | **Train\_Acc = 0.97 / Test\_Acc = 0.76**
- **1차 시도의 데이터보다 2차 시도에서 시행한 평준화 하지 않은 데이터가 더욱 높은 성능 수치를 보임**

## Model : KoBert 모델 학습 결과

KoBert 모델 성능표

번호	사용파일	사용모델	세부파라미터	train_test_split	GPU/CPU	리뷰 수	epochs	train_정확도	test_정확도
1	clean_total_crawling10.2 (1500ea even)	KOBERT	max_len = 64 batch_size = 64 warmup_ratio = 0.1 num_epochs = 3 max_grad_norm = 1 log_interval = 200 learning_rate = 5e-5	0.25	GPU	24785	3	0.23	0.20
2	clean_total_crawling10.2 (1500ea even)	KOBERT	max_len = 64 batch_size = 64 warmup_ratio = 0.1 num_epochs = 5 max_grad_norm = 1 log_interval = 200 learning_rate = 5e-5	0.25	GPU	24785	5	0.52	0.43
3	clean_total_crawling10.2 (1500ea even)	KOBERT	max_len = 64 batch_size = 64 warmup_ratio = 0.1 num_epochs = 10 max_grad_norm = 1 log_interval = 200 learning_rate = 5e-5	0.25	GPU	24785	10	0.65	0.60
4	clean_total_crawling10.2 (1500ea even)	KOBERT	max_len = 64 batch_size = 64 warmup_ratio = 0.1 num_epochs = 10 max_grad_norm = 1 log_interval = 200 learning_rate = 5e-5	0.25	TPU	24785	10	0.93	0.70
5	clean_total_crawling10.1	KOBERT	max_len = 64 batch_size = 64 warmup_ratio = 0.1 num_epochs = 5 max_grad_norm = 1 log_interval = 200 learning_rate = 5e-5	0.2	TPU	55042	5	0.87	0.73
6	clean_total_mecab	KOBERT	max_len = 64 batch_size = 64 warmup_ratio = 0.1 num_epochs = 20 max_grad_norm = 1 log_interval = 200 learning_rate = 5e-5	0.2	TPU	51034	20	-	-
7	clean_total_mecab	KOBERT	max_len = 64 batch_size = 64 warmup_ratio = 0.1 num_epochs = 2 max_grad_norm = 1 log_interval = 200 learning_rate = 5e-5	0.2	TPU	51034	2	0.74	0.74
8	clean_total_mecab	KOBERT	max_len = 64 batch_size = 64 warmup_ratio = 0.1 num_epochs = 1 max_grad_norm = 1 log_interval = 200 learning_rate = 5e-5	0.2	GPU	51034	1	-	-
9	clean_total_mecab	KOBERT	max_len = 64 batch_size = 64 warmup_ratio = 0.1 num_epochs = 20 max_grad_norm = 1 log_interval = 200 learning_rate = 5e-5	0.2	GPU	51034	20	0.97	0.76

### 최종 학습 결과

- 데이터 전처리와 형태소 분석을 통해 정제된 Data Set을 통해 최종적으로 **Train\_Acc = 0.97 Test\_Acc = 0.76**의 정확도 도출

### 성능 향상 원인

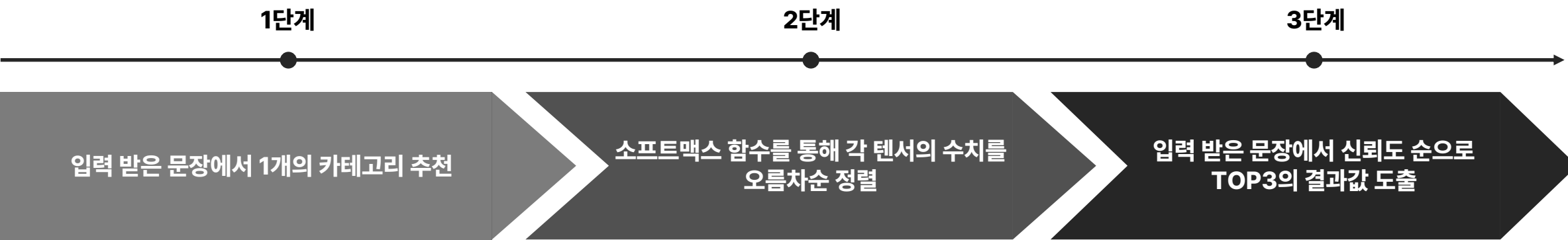
- 이상치의 카테고리를 제외하고 카테고리 별 상당수가 2500개 이상의 데이터 보유
- 카테고리 별 데이터 수를 1500개로 평준화 했을 때보다 **원본 데이터에 양질의 데이터가 더 많이 분포되어 있다고 판단**
- 때문에 데이터 불균형이 있더라도 양질의 학습 데이터 양이 많을수록 정확도 향상

### 모델 개선 방향

- 데이터 평균 개수를 1500개가 아닌 3000개 이상으로 맞출 경우 Test\_Acc의 개선이 대폭 올라가 **80~90% 수준으로 기대 및 판단**



| Model : KoBert 모델의 결과값 개선



<pre>[ ] predict('친구랑 축축하고 후루룩먹을 수 있는거 먹고싶어')</pre> <p>&gt;&gt; 입력하신 내용을 종합해 분석결과, 디저트(가) 종류의 음식들은 어떠신가요?</p>	<pre>[ ] testModel(model, '말랑하고 쫄쫄하면서 매콤한 음식이 땡기고 달짝지근한 음식 없을까')</pre> <pre>tensor([[ 0.6910, -0.5917, -0.7359, -0.7021, -1.8168, -1.1164, -1.2773,  9.9754,           -1.1736, -2.7342, -0.1401,  0.7301, -0.6098, -0.7408, -0.9994,  0.8516,            0.3699]], device='cuda:0', grad_fn=&lt;AddmmBackward0&gt;)</pre> <pre>torch.return_types.sort(   values=tensor([[3.0199e-06, 7.5586e-06, 1.2964e-05, 1.4380e-05, 1.5227e-05, 1.7117e-05,                   2.2169e-05, 2.2278e-05, 2.3043e-05, 2.5273e-05, 2.5735e-05, 4.0421e-05,                   6.7314e-05, 9.2804e-05, 9.6503e-05, 1.0897e-04, 9.9941e-01]],                   device='cuda:0', grad_fn=&lt;SortBackward0&gt;),   indices=tensor([[ 9,  4,  6,  8,  5, 14, 13,  2,  3, 12,  1, 10, 16,  0, 11, 15,  7]]),   device='cuda:0'))</pre> <p>추천 카테고리는: 분식 신뢰도는: 99.94%</p>	<pre>[ ] testModel(model, "분위기 좋고 아채없이 패티와 치즈만 선택해서 버거로 먹을 수 있는 것도 특이하고 무엇보다 맥앤치즈가 짱")</pre> <p>추천하는 TOP3 카테고리&amp;신뢰도:</p> <p>버거 99.94 %</p> <p>돈카츠 0.01 %</p> <p>베이커리 0.01 %</p> <hr/> <p>최종 분석결과 가장 적합한 음식은 '버거'이며, 문장 신뢰도는 99.94% 입니다.</p>
<pre>[ ] predict('입에 넣어서 씹자마자 육즙이 팡팡터지고 씹으면 씹을수록 고소함과 감칠맛을 느낄수 있는 음식이 먹고싶어요')</pre> <p>&gt;&gt; 입력하신 내용을 종합해 분석결과, 치킨(가) 종류의 음식들은 어떠신가요?</p>	<pre>[ ] testModel(model, '기름지고 마늘향이 나는 파스타가 먹고싶어')</pre> <pre>tensor([[ -1.2425, -2.1167, -1.5662, -0.3091, -1.4362,  0.7812,  1.0787, -0.9444,            0.1469, -0.3899,  9.7375, -1.2119, -1.6448, -0.7991, -0.0516, -2.2873,            0.6442]], device='cuda:0', grad_fn=&lt;AddmmBackward0&gt;)</pre> <pre>torch.return_types.sort(   values=tensor([[5.9896e-06, 7.1038e-06, 1.1388e-05, 1.2319e-05, 1.4029e-05, 1.7027e-05,                   1.7557e-05, 2.2941e-05, 2.6529e-05, 3.9941e-05, 4.3301e-05, 5.6018e-05,                   6.8322e-05, 1.1233e-04, 1.2883e-04, 1.7347e-04, 9.9924e-01]],                   device='cuda:0', grad_fn=&lt;SortBackward0&gt;),   indices=tensor([[15,  1, 12,  2,  4,  0, 11,  7, 13,  9,  3, 14,  8, 16,  5,  6, 10]]),   device='cuda:0'))</pre> <p>추천 카테고리는: 양식 신뢰도는: 99.92%</p>	<pre>[ ] testModel(model, "육즙이 팡팡 터지는 고소한 음식이 먹고싶어")</pre> <p>추천하는 TOP3 카테고리&amp;신뢰도:</p> <p>양식 99.85000000000001 %</p> <p>타코 0.05 %</p> <p>아시아음식 0.02 %</p> <hr/> <p>최종 분석결과 가장 적합한 음식은 '양식'이며, 문장 신뢰도는 99.85% 입니다.</p>
<pre>[ ] predict('어제 문제를 해결하지 못하여 몹시 지쳤을 때 기운을 낼 수 있을 정도로 든든하고 푸짐하며 얼큰한 음식을 먹고 싶어요')</pre> <p>&gt;&gt; 입력하신 내용을 종합해 분석결과, 스키(가) 종류의 음식들은 어떠신가요?</p>		
<pre>[ ] predict('시원하고 적당한 음식')</pre> <p>&gt;&gt; 입력하신 내용을 종합해 분석결과, 디저트(가) 종류의 음식들은 어떠신가요?</p>		
<pre>[ ] predict('다음 주에 진행할 프로젝트에서 떨지 않을 수 있을 정도로 감동적인 음식을 먹고 싶어요')</pre> <p>&gt;&gt; 입력하신 내용을 종합해 분석결과, 디저트(가) 종류의 음식들은 어떠신가요?</p>		

# part3. WEB

# Result

---

## | Web : 서비스 구현 과정

- 사용자 입장에서 편리하게 사용할 수 있도록 분석 모델을 시각적으로 보여줄 플랫폼이 필요하다고 판단



01

### Telegram Bot

- Telegram 앱에서 사용할 수 있는 자동화된 프로그램
- 사용자는 Telegram 앱을 통해서 봇에게 메시지를 보내거나 명령을 전송할 수 있음
- 봇은 사용자의 명령에 따라 적절한 작업 수행 가능



02

### Streamlit

- 웹 기반의 데이터 사이언스 및 머신러닝 애플리케이션을 만드는 프레임워크
- 사용자가 간단하게 대시보드, 분석 결과, 모델 시각화를 생성하여 손쉽게 웹에 게시하고 팀과 함께 공유 가능

## Streamlit 선정 이유

- 초기 프로젝트 모델은 챗봇이었기 때문에 Telegram Bot이 언급되었으나, 프로젝트의 방향성이 고도화 되면서 대화 형식의 챗봇보다 웹 기반 서비스 선택
- Streamlit 의 여러 기능을 사용하여 Map / DataFrame 등의 다양한 시각화 정보를 제공

# Result

## | Web : 서비스 구현 과정 - Main

INPUT

search map

🗨️ 오늘도 무엇을 먹을지 고민하고 계신가요?

지금 생각나는 키워드를 입력하고 Ctrl+Enter를 눌러주세요!

Ex) 육즙이 팡팡 터지는 고소한 음식이 먹고싶어.

### 1 Tab 기능

Streamlit의 Tab기능을 통해 Search와 Map으로 나눠 2개의 Page 생성

### 2 Main Service

- 메인 KoBert 분류 모델 적용
- 사용자가 직관적으로 우리의 서비스를 이해할 수 있게끔 친숙한 문장으로 Headline 작성
- Sub Headline 에 추가적인 도움말 제시
- 사용자의 Text Input Box 구현 : 문장 작성에 제한이 없도록 크기 조절이 가능한 Text Box 생성
- Placeholder 속성을 이용하여 사용자가 적절한 값을 입력할 수 있도록 도와주는 도움말 제시

### OUTPUT

search map

🗨️ 오늘도 무엇을 먹을지 고민하고 계신가요?

지금 생각나는 키워드를 입력하고 Ctrl+Enter를 눌러주세요!

어제 술을 마셨더니 해장하고 싶어. 매콤하고 칼칼한 음식 추천해줘!

### 한식

한식이(가) 가장 적합한 음식입니다. 신뢰도는 88.5% 입니다.

입력문장과 가장 일치하는 음식 TOP3 입니다.

🍽️ 한식 신뢰도는 88.5 % 입니다.

🍴 분식 신뢰도는 11.14 % 입니다.

🍱 해산물 신뢰도는 0.17 % 입니다.

### 1 결과값 제공

- 학습한 KoBert 모델에 입력 받은 자연어를 처리한 후 신뢰도 순으로 결과값 제공
- 사용자가 더 많은 카테고리 내에서 결정할 수 있게끔 3가지 결과값 제시

### MAP INPUT

search **map**

#### 가시려는 지역이 어디인가요?

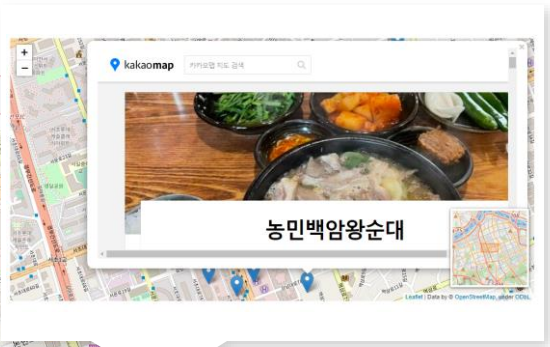
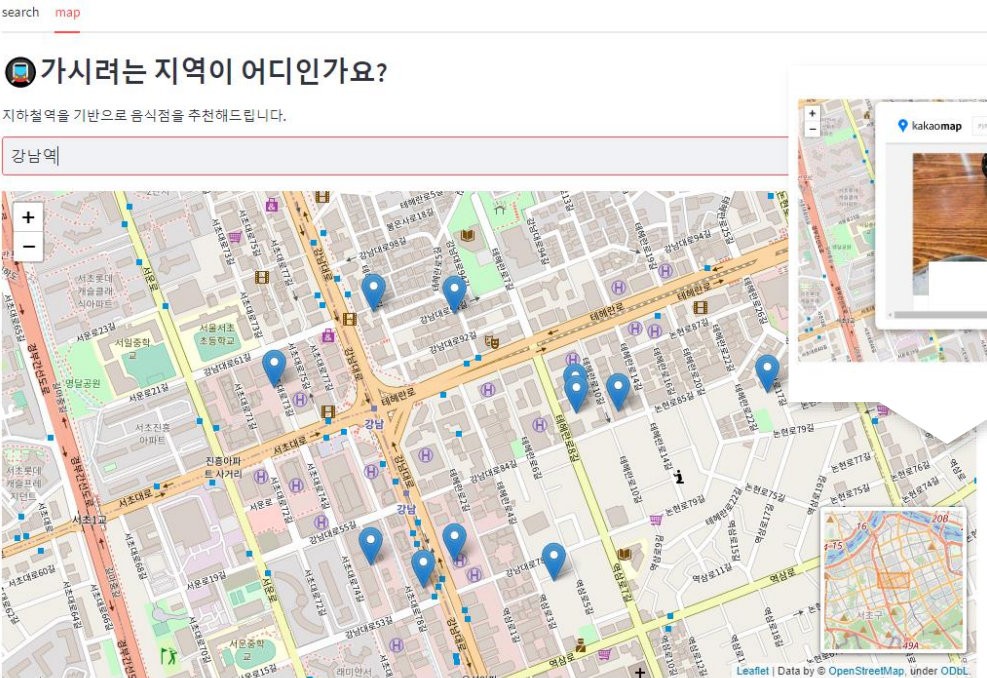
지하철역을 기반으로 음식점을 추천해드립니다.

근처 지하철 역을 입력해주세요. ex) 강남역

#### 1 Sub Service

- Search Page에서 사용자가 입력한 문장을 토대로 도출한 결과값을 저장
- 사용자가 이해하기 쉽도록 친숙한 문장으로 Headline 작성
- '지하철 역' 기반의 서비스임을 알리도록 자세한 Sub Headline 작성
- 사용자가 음식을 먹을 수 있는 지역을 입력하도록 Text Box 생성
- Placeholder 속성을 이용하여 사용자가 적절한 값을 입력하도록 도움말 제시

MAP OUTPUT



	stores	road_address	phone	place_url
0	농민백암왕순대	서울 강남구 역삼로3길 20-4	02-501-2772	http://place.map.kakao.com/16421356
1	창고43 강남점	서울 강남구 강남대로 362	02-508-0043	http://place.map.kakao.com/1503746075
2	강남진해장	서울 강남구 테헤란로5길 11	02-557-2662	http://place.map.kakao.com/1428481536
3	장인닭갈비 강남점	서울 강남구 테헤란로1길 19	02-3452-3441	http://place.map.kakao.com/95713992
4	두껍삼 역삼직영점	서울 강남구 테헤란로8길 17	1670-3592	http://place.map.kakao.com/2011092566
5	돌고기506	서울 강남구 역삼로17길 53	02-6933-9501	http://place.map.kakao.com/1511737742
6	노랑저고리 강남점	서울 서초구 서초대로73길 9	02-534-5300	http://place.map.kakao.com/22716674
7	서관면옥	서울 서초구 서초대로56길 11	02-521-9945	http://place.map.kakao.com/34321780
8	신동궁감자탕 역삼본점	서울 강남구 테헤란로10길 21		http://place.map.kakao.com/7983060
9	대우부대찌개	서울 강남구 테헤란로25길 34	02-552-1663	http://place.map.kakao.com/7819490

결과는 인기도순으로 반영되었습니다.

1 카카오맵 API 연동

- 카카오맵 API를 연동하여 사용자가 입력한 지하철 역을 기반으로 인기도순 식당 LIST 제공
- '지하철역' + '결과값' (ex. '강남역 한식')의 형태로 카카오맵 검색 자동화
- Folium 라이브러리를 이용하여 Marker 표시 : 클릭 시 식당의 추가적인 정보 제공
- 사용자가 한눈에 보기 쉽게 지도에 표시된 식당을 DataFrame화 하여 정보를 쉽게 파악할 수 있도록 함

# **part4. NEXT STEP**





| Next Step : 서비스의 차별성 1

Google

영등포역에서 여자친구랑 갈만한 음식점

전체

지도

이미지

뉴스

동영상

더보기

도구

검색결과 약 491,000개 (0.44초)

fmkorea.com

https://www.fmkorea.com > ...

영산/홍대/영등포 여자친구랑 100일때 갈만한 음식점 있을까요?

2022. 12. 4. — 예산은 총 5~6정도 보고있어요여자친구 흥강하고 오마카세 가기로 해서 ... 둘다 대학생이라 크게 우리는 안하려고요~적당히 분위기 있는 맛집 있을까요?

https://www.fmkorea.com > ...

홍대/영산/영등포 여친이랑 갈만한 맛집 추천 좀 ㅇㅇ - 에phem코리아

2022. 12. 4. — 홍대/영산/영등포 여친이랑 갈만한 맛집 추천 좀 ㅇㅇ. [레벨:24] 김용수 ... 신범수 보통 타임스퀘어에서 만나는데 근처면 상관없으. [레벨:2] ...

pgr21.com

https://pgr21.com > qna

[질문] 신도림, 영등포, 구로역 쪽에 기념일에 갈만한 ... - Pgr21

2017. 5. 7. — 아무튼 각설하고 여친과의 기념일에 갈만한 식당 중 신도림, 영등포, 구로역 쪽에서 추천 부탁 드립니다. 저희는 20대 후반이며 예산은 인당 10만원 ...

siksinhot.com

https://www.siksinhot.com > ...

25센치꼬치엔오뎅바 영등포직영점 - 식신

맛집 25센치꼬치엔오뎅바 영등포직영점은 6종 꼬치세트,꼬꼬덕세트,파인애플 샤베트,메론반통을 잘하는 베스트 맛집으로 회식, 모임, 데이트, 주전맛집 입니다.

누락된 검색어: 갈만한 | 다음 정보가 포함되어야 합니다. **갈만한**

naver.com

https://m.blog.naver.com > phjsunflower

영등포역 맛집 / 핫플 청기와타운 본점 - 네이버 블로그

2021. 6. 25. — 주문을 하면 반찬이 푸욱 깔린다. 계장부터 시작을 해서 각종소스를 무채와 샐러드 고구마무스까지. 정갈한 한상이 차려지는데 꽤 예쁘게 플레이팅이 ...

N

영등포역에서 여자친구랑 갈만한 음식점

통합

VIEW

이미지

지식IN

인플루언서

동영상

쇼핑

뉴스

어학사전

지도

...

VIEW

전체

블로그

카페

뱅글뱅글 라이프 \*~\* | 2023.01.04.

[영등포구/영등포역] 정통집 영등포점 / 북적북적한 돼지김치구이...  
여자 2명이 먹는데 중짜를 시켜서 은근 염려가 되었으나. 고맙겠다! 말끔하게 싹싹 다 먹을 수 있다. 2명이면... 이 음식점을 소개해 준 친구에게 고맙다고 말했다! 다...

15

호기심 가득 민장의 도전일기 | 2023.02.07.

[영등포역 술집/호삼촌] 안주 맛집 분위기 좋은 신상술집  
건질만한 사진 많은 여보슈 다들 한번씩 전화부스 앞에서 사진 찍길래 괜히 나도 한 번 찍어봄...ㅎㅎ 생각보다... 남자친구랑 술집을 방문했는데 분위기도 좋고 안주도 ...

35

레미로그 | 인플루언서 | 2일 전

영등포역 맛집 화양연화 타임스퀘어점 쌀국수 성공적  
광명 갈 일이 생겨 지나가는 길에 영등포 사는 친구와 타임스퀘어에서 쌀국수를 먹기로 했어요! 사실 이번... 태국 음식점이기 때문에 질리기는커녕 또 새롭더라고요 ...

16

실버페이퍼's 이야기 | 2022.07.25.

영등포역 아시아 음식 맛집 피에프창 타임스퀘어점에 다녀왔어요 ~  
친절했고 음식점에서 디저트까지 해결할 수 있는 점이 무엇보다 너무 좋았어요 ~ 저는 다음번에도 타임스퀘어에서 약속이 있다면 재방문 의사 있습니다 ! 히히 여...

25

오늘도 무엇을 먹을지 고민하고 계신가요?

지금 생각나는 키워드를 입력하고 Ctrl+Enter를 눌러주세요!

여자친구랑 데이트할 때 먹을만한 음식

베이커리

베이커리(가) 가장 적합한 음식입니다. 신뢰도는 71.51% 입니다.

입력문장과 가장 일치하는 음식 TOP3 입니다.

베이커리 신뢰도는 71.51 % 입니다.

해산물 신뢰도는 15.63 % 입니다.

디저트 신뢰도는 8.75 % 입니다.

가시려는 지역이 어디인가요?

지하철역을 기반으로 음식점을 추천해드립니다.

영등포역



구글과 네이버 검색 엔진에서 비슷한 문장을 입력했을 때, 다양한 정보 때문에 또 다시 선택에 어려움을 겪는 문제가 발생  
2거 어때 모델은 확실한 카테고리 추천을 통해 선택지를 좁혀주는 점에서 다른 검색 엔진과의 차별성을 가지고 있음

# Result

## | Next Step : 서비스의 차별성 2

소연

친구에게 햄치미를 물어보면서 먹을 수 있는 음식 추천해줘

안녕하세요! 친구가 햄치즈를 좋아하는지 물어보고 음식을 추천해 드릴게요.

1. 햄치즈 피자: 햄치즈가 고소하게 구워진 피자는 절대 놓치지 마세요!

2. 햄치즈 베이글: 토스트된 베이글 위에 햄치즈와 올리브 오일을 곁들인 것은 맛있고 간단한 식습관입니다.

3. 햄치즈 샌드위치: 햄, 야채, 마요네즈와 같은 재료를 사용하여 만든 샌드위치는 햄치즈의 맛을 최대한 느낄 수 있습니다.

4. 햄치즈 스팸 토스트: 스팸과 햄치즈, 여러 가지 야채를 사용하여 만든 토스트는 좋은 아침 식습관입니다.

5. 햄치즈 오믈렛: 햄치즈, 계란, 마요네즈 등의 재료를 사용하여 만든 오믈렛은 간식용으로 좋습니다.

이상 5가지 추천이에요! 즐겁게 맛있는 음식

🗨️오늘도 무엇을 먹을지 고민하고 계신가요?

지금 생각나는 키워드를 입력하고 Ctrl+Enter를 눌러주세요!

친구에게 햄치미를 물어보면서 먹을 수 있는 음식 추천해줘

버거

버거이(가) 가장 적합한 음식입니다. 신뢰도는 99.96% 입니다.

입력문장과 가장 일치하는 음식 TOP3 입니다.

🍔 버거 신뢰도는 99.96 % 입니다.

🍷 디저트 신뢰도는 0.01 % 입니다.

🍰 베이커리 신뢰도는 0.01 % 입니다.

📍가시려는 지역이 어디인가요?

지하철역을 기반으로 음식점을 추천해드립니다.

강남역



Chat GPT와 2거 어때 모델에 똑같은 문장을 입력했을 때 답변 차이로 보아,  
음식과 식당 추천 분야에서는 2거 어때 모델이 Chat GPT의 결과값보다 우수

- 신뢰도
- 지도 정보 제공
- 신조어 학습

위 3가지 키워드에 대해서는 Chat GPT보다 2거 어때 모델이 적합하다고 판단

# Result

---

| Next Step : 향후 개선 방향



## Data

확대적 측면

**다량의 리뷰 데이터 추가 수집**

데이터 성능 향상

**카테고리 확대**

더 많은 양질의 결과값 제공

## Business

비즈니스적 측면

**배달의 민족 / 요기요 등의 배달 플랫폼과의 협업**

배달의 민족 / 요기요 등의 메인 서비스가 아닌 부가적인 서비스로 음식 선택을 도와주는 탭을 제공

**웹 무료 배포**

다수의 사람들에게 양질의 서비스를 제공

# Result

## I 느낀점



홍세영

대학 때 경험했던 팀 프로젝트와는 다르게 멀티캠퍼스를 통해 참여하게 된 이 프로젝트를 통해 느낀 점이 있다면 더욱 겸손해야겠다는 것이었습니다. 처음에는 기술을 잘 이해하는 것만 중요하다고 생각했지만, 아무리 유능 해도 결국 사회에서는 협업을 해야 하기 때문에 소통 능력도 기술을 이해하는 만큼 중요하다는 것을 프로젝트를 통해 느껴볼 수 있었습니다. 사회로 나가기 전에 재미있는 경험을 할 수 있도록 노력해준 팀원들과 강사님께 감사의 인사를 드립니다. (31"~")

메인프로젝트를 진행하면서 막힐 때마다 주제에 대한 회의감이 들었는데 끝까지 포기하지 않고 결국은 원하는 결과를 만들어낸 2조 팀원 모두에게 고생했다 말씀드리고 싶고 감사했습니다! ★ (31~5~) ★



이준석



김소연

1달 반이라는 긴 시간 동안 프로젝트를 진행하면서 팀원들과 의견을 조율하는 부분도 어려웠고 생각한대로 결과물이 나오지 않아 스트레스도 많이 받았는데 결국에는 원하는 결과를 얻어내서 뿌듯했습니다. 서비스를 구현하는 과정에서 웹이나 서버 지식을 더 공부했더라면 훨씬 완성도 높은 결과물을 구현했을텐데 그러지 못한 게 아쉬웠습니다. 9월부터 달려온 여러분들과 강사님, 그리고 1달 반 동안 고생해주신 파이널 팀원 분들께 가장 감사드리며 앞으로 좋은 일만 가득하셨으면 좋겠습니다!! ♥ (31~5~)

시작할 당시만해도 결과가 나올까 걱정하던 팀원들이 이제는 꽤 그럴듯하다 하는 것을 보고 보람차다고 느꼈습니다. 사실상 모두가 처음인 프로젝트에서 원하는 방향성까지 시도하는 과정은 성공보다는 실패가 많았습니다. 하지만 실패를 쌓아 성공할 수 있게 되어 더욱 뿌듯하게 느껴지기에 성공과 실패의 유무를 떠나 저희 팀은 성공했다고 생각합니다. 팀원 분들께 고생했다는 말을 전하고 싶습니다. (b`<` )b



진세용



# 감사합니다

