

# 딥 러닝을 활용한 배달 후기 감정 분석 및 단체 배달 간소화 서비스

---

팀명 : 도 레 미 파 솔

팀장 : 최 상문

팀원 : 김 민수 이 한주 서 주연 조 혜원



## 발표 순서

---

1

제안 배경

2

개발 과정

- WEB 기능 구현
- 딥러닝 분석
- 머신러닝 분석

3

시연 영상

4

기대효과 & 활용방안

5

일정 & 개발 툴

6

팀원 소개

7

Q & A



# 제안배경

---





# 팀원 사전 면담

Cat.	List	최상문	김민수	조혜원	이한주	서주연
Personal	전공, 비전자	전공+비전공	비전공자(법학과)	전공자(컴퓨터공학과)	비전공자(공대생)	비전공자
	고향, (거주지 근처 건물)	포항, 호주, 풍암	순천, 광주역근처	광주, 버스 터미널	첨단 2지구	광주 토박이, 진월동
	업무 경험	O	X	O	X	O
	일할때 성격	싸우고, 격없는 논쟁 선호	임기응변, 의사소통	시켜주는거	분위기에 따라 달라짐	일 시켜주는거 좋아함
	희망 역할	필요한 역할	백엔드&응용	프론트	발표 제외, 전부	프론트
Experience	1차 프로젝트 역할	팀장	팀원	팀원	팀원	팀원
	1차 프로젝트 후회	팀원 관리 실패	시키는 것만 한 점	각자의 역할만 했던 점	내 역할만 한 것	개인적으로 갔던 거
	2차 프로젝트 역할	팀원	팀장	팀장	팀원	팀원
	2차 프로젝트 후회	역할을 더 맡아서 할걸....	리딩 디테일 개발	팀장으로서 많이 부족함을 느꼈	의견을 제대로 안낸 것	내 의견을 제대로 안낸 것
	3차 프로젝트 바라는점	팀원과의 대화	의사소통	팀워크	무사히 프로젝트 완료	화기애애한 분위기
	3차 프로젝트 역할	팀장	팀원	팀원	팀원	팀원
	3차 프로젝트 목표	팀원들과 후회없는 프로젝트	역할 수행을 통한 실력 향상	프로젝트 1등!	적극적인 프로젝트 참여	프로젝트 목표 완벽 구현
Skill	DAO VO	★	★	★★★	★	★★★★★
	JSP, Servlet	★	★	★★★	★	★★★★★
	DB	★★	★★	★★	★★★	★★★★★
	Query	★	★★	★★	★★★	★★★★★
	HTML	★	★★★★	★★★★	★	★★
	CSS	★	★★★★	★★★★	★	★★
	파이썬	★★★★	★★★★	★★★	★★★★	★★★★
	크롤링	★★★★	★★★★	★★★	★★★★	★★★★
	머신러닝	★★★★	★★★★	★★	★★★	★★★★
	딥러닝	★★★★★	★★★★	★★	★★★	★★★★
Wish	팀명		TEAM SangMoon_Choi	Super moon(엄청 큰 달) 최상문		Super moon(엄청 큰 달)
	팀장한테 바라는거		의사소통	어느 하나에 집착 금지	없음	없음
	프로젝트 아이디어			구글 문서 공유 링크 : Link		
	요즘 걱정?		진로, 가족	진로	진로	진로.
	프로젝트 철학	구성원 적극적 참여	기발하거나, 일상적이거나 공감이 되는 주제	사용에 용이한것	너무 필요한데 없는 것	



# 팀원 사전 면담

## 프로젝트 목표 (철학)

프로젝트에 **공감대**가 없다면  
열정을 쏟을 수 없다 !

기술을 보여주는 개발보다  
**사람들이 원하는 서비스를 개발**해보자

Cat.	List	최상문	김민수	조혜원	이한주	서주연
Personal	전공, 비전자	전공+비전공	비전공자(법학과)	전공자(컴퓨터공학과)	비전공자(공대생)	비전공자
	고향, (거주지 근처 건물)	포항, 호주, 풍암	순천, 광주역근처	광주, 버스 터미널	첨단 2지구	광주 토박이, 진월동
	업무 경험	O	X	O	X	O
	일할때 성격	싸우고, 격없는 논쟁 선호	임기응변, 의사소통	시켜주는거	분위기에 따라 달라짐	일 시켜주는거 좋아함
	희망 역할	필요한 역할	백엔드&응용	프론트	발표 제외, 전부	프론트
Experience	1차 프로젝트 역할	팀장	팀원	팀원	팀원	팀원
	1차 프로젝트 후회	팀원 관리 실패	시키는 것만 한 점	각자의 역할만 했던 점	내 역할만 한 것	개인적으로 갔던 거
	2차 프로젝트 역할	팀원	팀장	팀장	팀원	팀원
	2차 프로젝트 후회	역할을 더 맡아서 할걸....	리딩 디테일 개발	팀장으로서 많이 부족함을 느꼈	의견을 제대로 안낸 것	내 의견을 제대로 안낸 것
	3차 프로젝트 바라는점	팀원과의 대화	의사소통	팀워크	무사히 프로젝트 완료	화기애애한 분위기
	3차 프로젝트 역할	팀장	팀원	팀원	팀원	팀원
	3차 프로젝트 목표	팀원들과 후회없는 프로젝트 진행	역할 수행을 통한 실력 향상	프로젝트 1등!	적극적인 프로젝트 참여	프로젝트 목표 완벽 구현
Skill	DAO VO	★	★	★★★	★	★★★★★
	JSP, Servlet	★	★	★★★	★	★★★★★
	DB	★★	★★	★★	★★★	★★★★★
	Query	★	★★	★★	★★★	★★★★★
	HTML	★	★★★★	★★★★	★	★★
	CSS	★	★★★★	★★★	★	★★
	파이썬	★★★★	★★★★	★★★	★★★★	★★★★
	크롤링	★★★★	★★★★	★★★	★★★★	★★★★
	머신러닝	★★★★	★★★★	★★	★★★	★★★★
	딥러닝	★★★★★	★★★★	★★	★★★	★★★★
Wish	팀명		TEAM SangMoon_Choi	Super moon(엄청 큰 달) 최상문		Super moon(엄청 큰 달)
	팀장한테 바라는거		의사소통	어느 하나에 집착 금지	없음	없음
	프로젝트 아이디어			구글 문서 공유 링크 : <a href="#">Link</a>		
	요즘 걱정?		진로, 가족	진로	진로	진로.
	프로젝트 철학	구성원 적극적 참여	기발하거나, 일상적이거나 공감이가 되는 주제		사용에 용이한것	너무 필요한데 없는 것





## Profile

이름 : 김민수  
나이 : 20대 후반  
성별 : 남성

순천 출신, 광주 거주중  
개발자 지망생  
비전공자

화목한 가정  
걱정없이 자란 외동아들

## Detail, Behavior

20대 비전공자이지만 개발자를 목표로 SMHRD 빅데이터 반에서 수업중이다.  
순천 출신으로 타지에 적응이 어렵다.

광주에서 IT 분야의 직장을 얻고 정착하기위해 실력을 늘리고 싶다.

지난 5월 개강 이후, 매일 점심때마다 학생들의 단체 배달 주문을 도맡아 처리하고 있다.

매일 점심마다 반복되는 메뉴 선정, 배달 픽업, 더치 페이 등의 의도치 않은 불편으로 3개월째 스트레스를 받고 있다.

## Problems

- 매일 15명에 가까운 점심 주문을 책임진다.
- 얻는 이득 하나 없이 책임만 늘어난다.
- 배달 음식 위주로 식사로 살이 찌고 있다.
- 수업에만 집중하고 싶다.
- 프로그래밍 실력이 향상되고 싶다.
- 불규칙적인 생활을 하고있다.

## Needs

- 다른 사람이 점심 주문을 했으면 좋겠다.
- 단체 주문 과정이 간단했으면 좋겠다.
- 15명의 결제 금액이 부담된다.
- 메뉴마다 다른 15명의 더치페이 금액 계산을 쉽게 하고싶다.
- 메뉴 선택이 쉬웠으면 좋겠다.
- 자신이 주문했으면 바로 입금해줬으면 좋겠다.
- 입금을 꾀박한 사람에게 독촉하기 싫다.





## Profile

이름 : 김민수  
나이 : 20대 후반  
성별 : 남성

순천 출신, 광주 거주중  
개발자 지망생  
비전공자

화목한 가정  
걱정없이 자란 외동아들

## Detail, Behavior

20대 비전공자이지만 개발자를 목표로 SMHRD 빅데이터 반에서 수업중이다.  
순천 출신으로 타지에 적응이 어렵다.

광주에서 IT 분야의 직장을 얻고 정착하기위해 실력을 늘리고 싶다.

지난 5월 개강 이후, 매일 점심때마다 학생들의 단체 배달 주문을 도맡아 처리하고 있다.

매일 점심마다 반복되는 메뉴 선정, 배달 픽업, 더치 페이 등의 의도치 않은 불편으로 3개월째 스트레스를 받고 있다.

## Problems

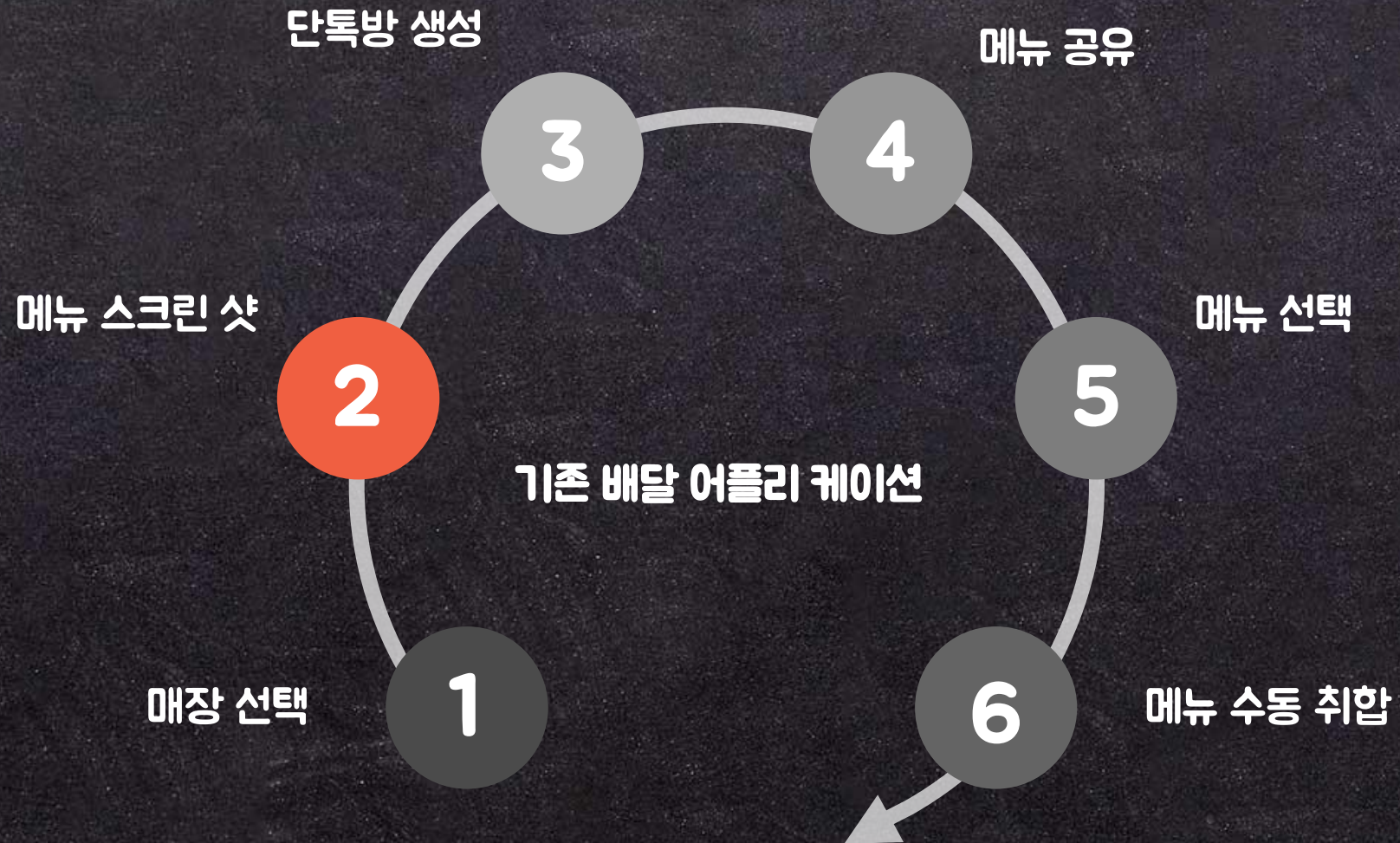
- 매일 15명에 가까운 점심 주문을 책임진다.
- 얻는 이득 하나 없이 책임만 늘어난다.
- 배달 음식 위주로 식사로 살이 찌고 있다.
- 수업에만 집중하고 싶다.
- 프로그래밍 실력이 향상되고 싶다.
- 불규칙적인 생활을 하고있다.

## Needs

- 다른 사람이 점심 주문을 했으면 좋겠다.
- 단체 주문 과정이 간단했으면 좋겠다.
- 15명의 결제 금액이 부담된다.
- 메뉴마다 다른 15명의 더치페이 금액 계산을 쉽게 하고싶다.
- 메뉴 선택이 쉬웠으면 좋겠다.
- 자신이 주문했으면 바로 입금해줬으면 좋겠다.
- 입금을 꾀박한 사람에게 독촉하기 싫다.



## 단체 배달 과정





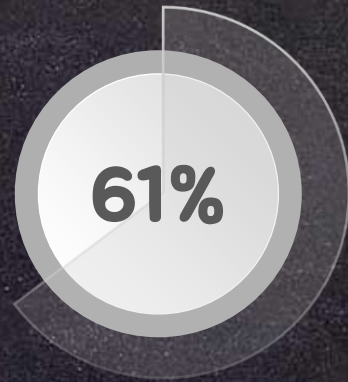
## 단체 배달 과정



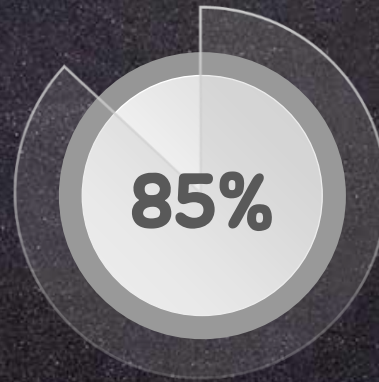


# 설문조사

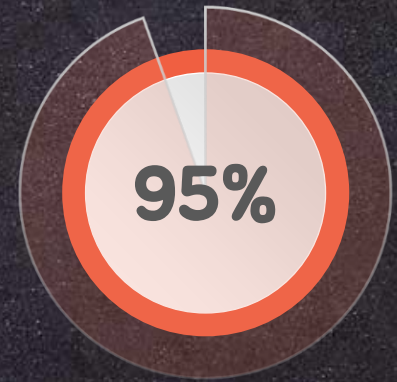
스마트 미디어 8개반 209명 중 84명이 응답



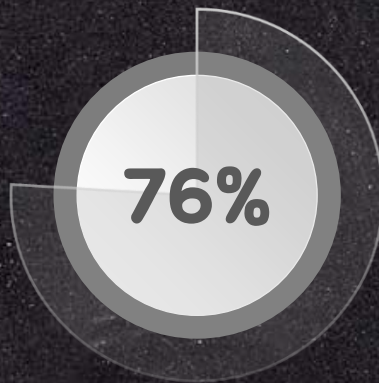
평균 단체 주문 횟수  
주 3회 이상



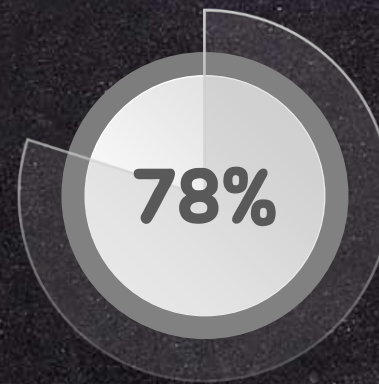
단체 주문 책임 맡기 싫음  
너무 복잡하다 58명(69%)



간편 단체주문 필요성  
필요하다(80명)

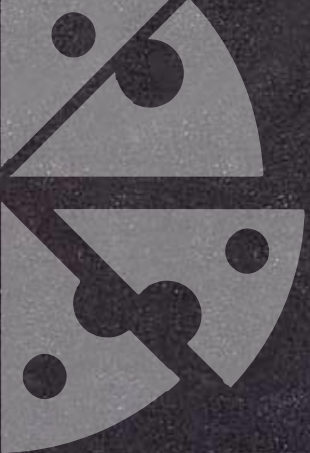


주문 책임 경험  
64명



불편함을 알고 있음  
65명







## 유사 서비스 현황

### 요기요 기업 주문 서비스

- 임직원이 등록 시 사용 가능
- 일반인 사용 불가



## 간편한 단체 주문의 부재



집에서도, 회사에서도 요기요 앱 하나로  
**기업 주문 서비스 오픈!**



**직원 여러분!**

회의 중 커피 주문도,  
회사 행사 단체 주문도  
요기요로 편리하게 하세요!



**관리팀 여러분!**

식대 관리의 피곤함에서 해방되세요!  
정산도, 새로운 음식점 섭외도  
요기요가 책임집니다.



**경영진 여러분!**

식대 지출 내역의 투명성을  
확보하세요! 선도적인 프로세스  
도입으로 근무환경이 향상됩니다.



# 배달 시장의 확대

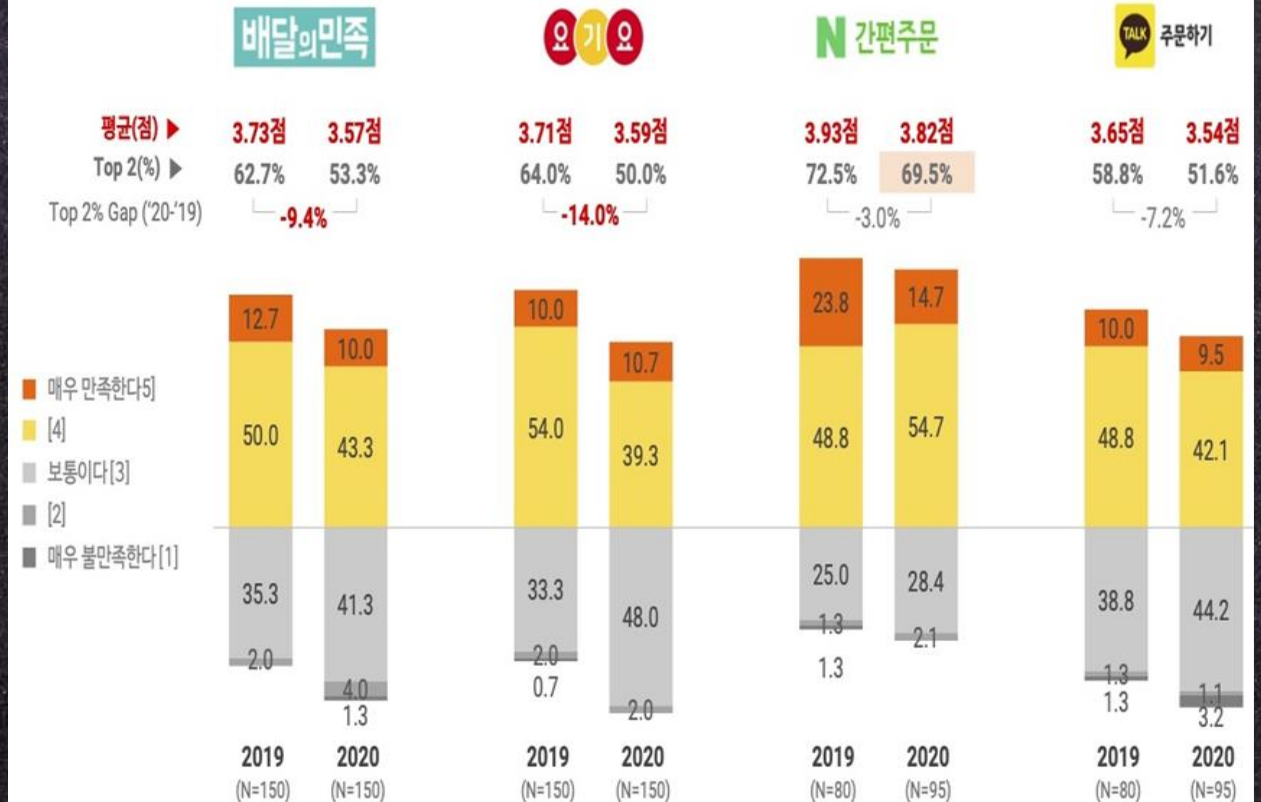
## 배달 서비스 만족도 하락

후기/리뷰 (신뢰) 를 장점으로 꼽는  
비율이 많이 감소했습니다.

- 배달의 민족 (17.3%, -7.4%p)
- 요기요 (16.0%, -9.3%)

실제로 배달 앱의 후기/리뷰 영역이  
마케팅 채널로 활용되면서 배달 앱에  
입점한 음식점들이 허위 리뷰를  
작성하는 등 사례가 발생하고 있습니다

배달 웹사이트/앱 서비스 별 만족도



[Base: 최근 3개월 내 각 배달 웹사이트/앱 서비스 이용 경험이 있는 사람, 척도형 응답, %]

[빨강 글꼴: 전년 대비 -8%p 차이 나는 데이터를 표시]



# 개발 방향

## 리뷰, 별점 신뢰도 하락

리뷰를 꼭 확인 (42%)  
더 이상 리뷰를 믿지 못함 (42%)  
배민, 요기요 신뢰도(-7.4%, -9.3%)

## 메뉴 선정의 어려움

응답자의 51%가 무엇을 먹어야 할지 모르겠다고 응답



## 단체 배달 주문 서비스 필요

대다수가 단체 배달의 불편함을 인지하고 있고,  
단체 주문 서비스의 간편화 필요성에 공감하고 있다.

## 단체 주문 수요 증가

코로나로 인해 단체 주문 수요가 증가함에도 시장에  
대체재가 없음



# 개발과정

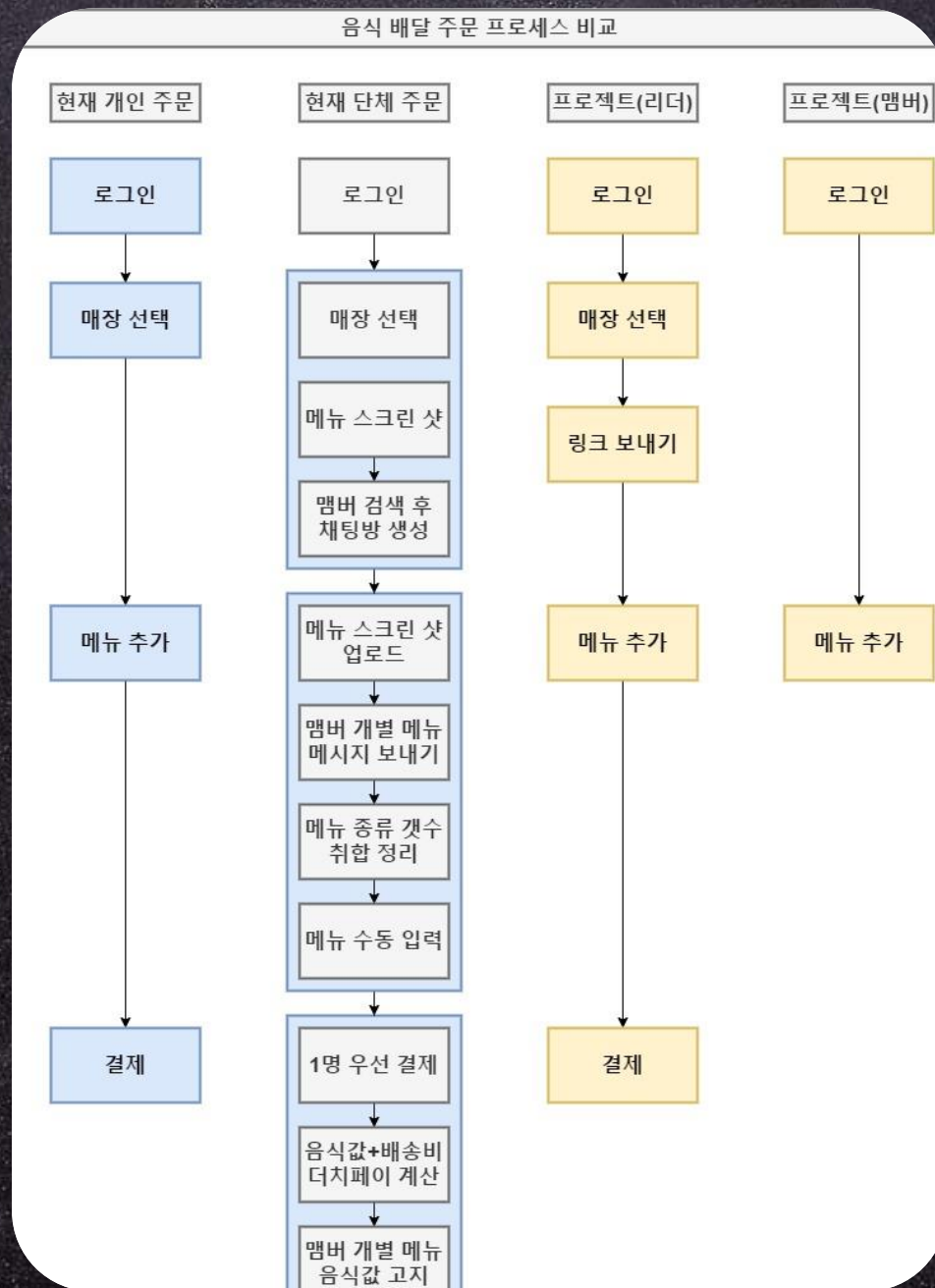
---





# FLOW CHART

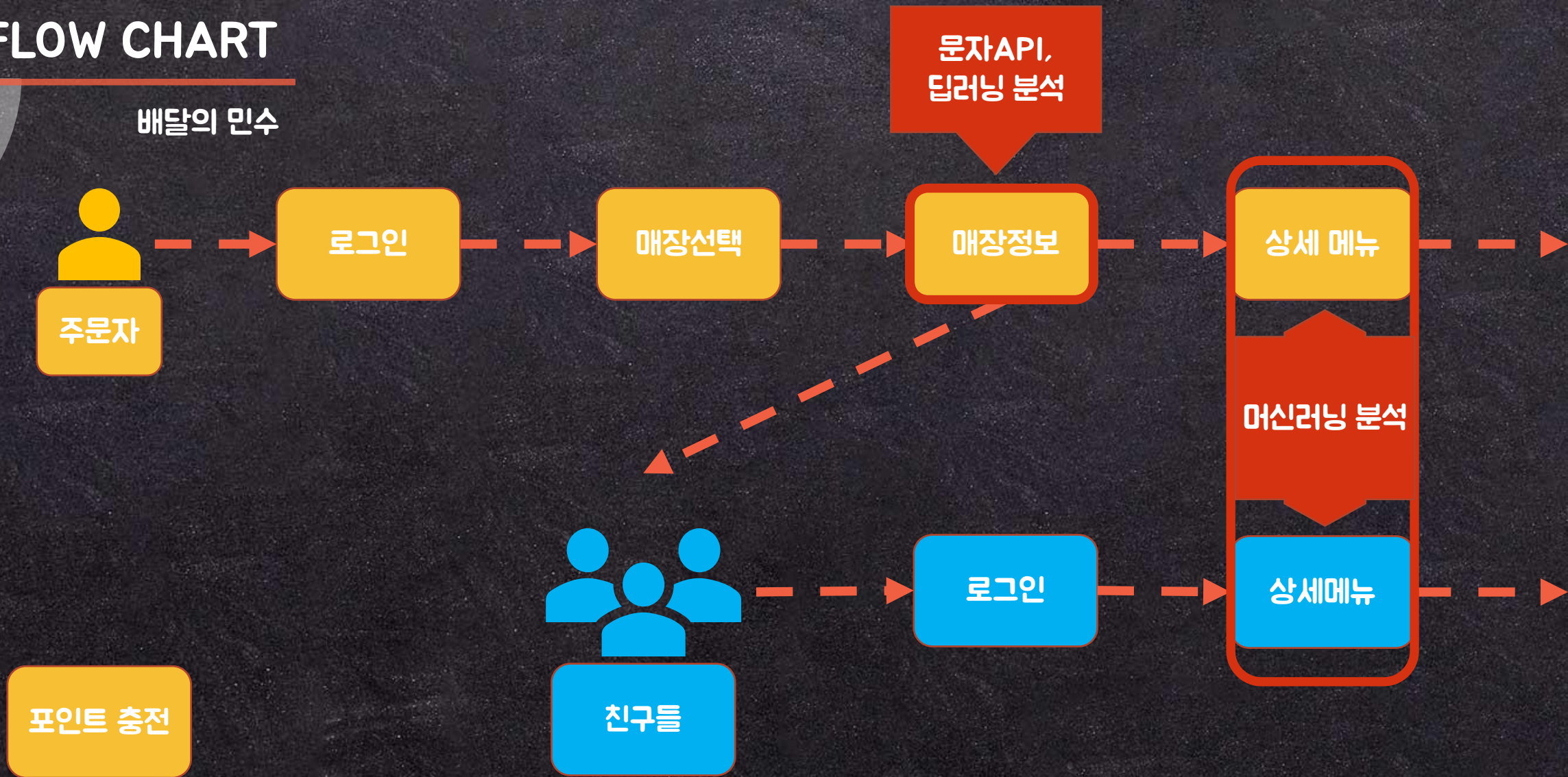
## 기존의 어플리케이션





# FLOW CHART

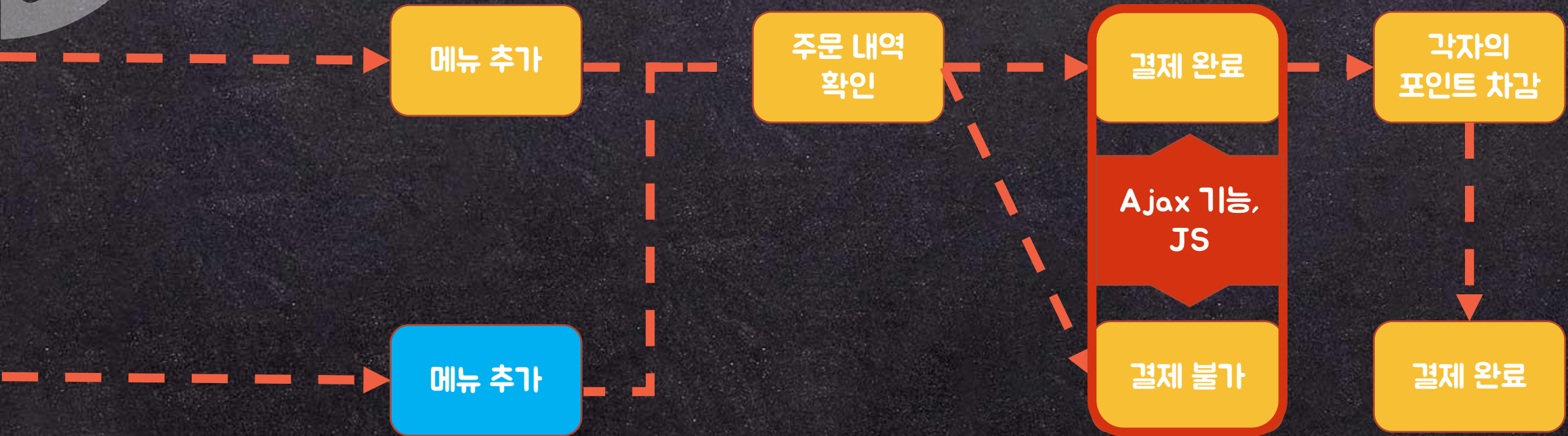
배달의 민수





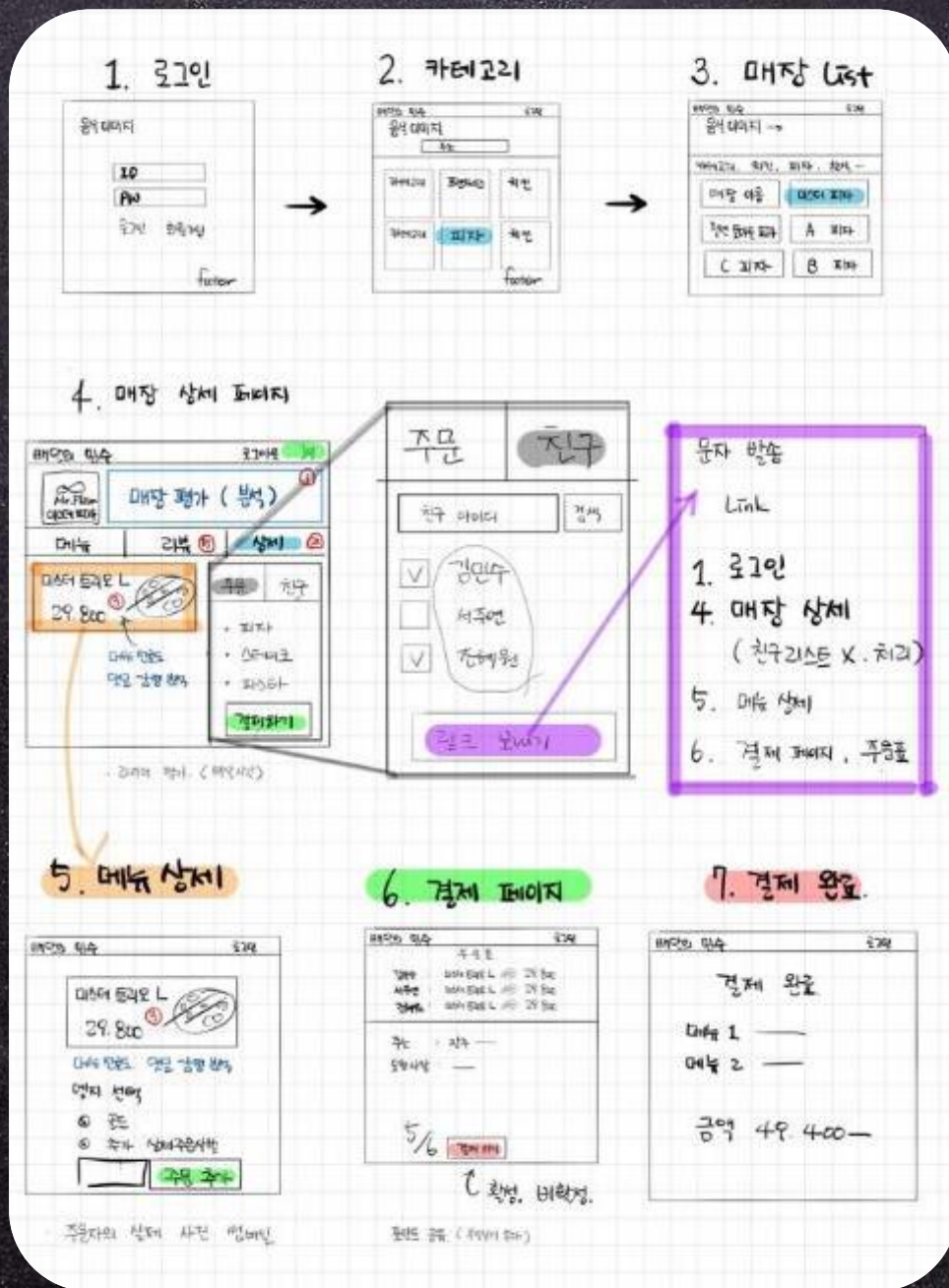
# FLOW CHART

배달의 민수





# Web Design 설계





# WEB 기능 구현

## 문자 서비스

```
request.setCharacterEncoding("euc-kr");
HttpSession session = request.getSession();
String[] invite_id = request.getParameterValues("invite_id");
String restaurant_id = (String)session.getAttribute("restaurant_id");
userVO vo = (userVO)session.getAttribute("vo");

session.setAttribute("cnt", invite_id.length+1);
String url = "http://127.0.0.1:8077/delivery_minsu_project/login2.jsp?restaurant_id=" + restaurant_id + "&code_id=" + vo.getId();

String api_key = "NC35APDLEIKHAK2";
String api_secret = "4SVX07UGAER7VST04NXTPMQ2TMMJE2RL";
Message coolsms = new Message(api_key, api_secret);

ArrayList<String> phone = new ArrayList<String>();
for (int i = 0; i < invite_id.length; i++) {
    phone.add(invite_id[i]);
}

if (phone.size() > 0) {
    for (int i = 0; i < phone.size(); i++) {
        HashMap<String, String> params = new HashMap<String, String>();
        params.put("to", phone.get(i));
        params.put("from", "01023998278");
        params.put("type", "SMS");
        params.put("text", url);
        params.put("app_version", "test app 1.2");

        try {
            JSONObject obj = (JSONObject) coolsms.send(params);
            System.out.println(obj.toString());
        } catch (CoolsmsException e) {
        }
    }
}
phone.clear();

return "menuchoice.jsp";
```

활용 기능 :

## 문자 API (COOL SMS)



주문공유 버튼 클릭 전 주문자가 체크한 친구들의  
전화번호로 식당아이디와 코드아이디를 메뉴페이지  
주소 뒤에 쿼리스트링으로 붙여 **url주소**를 생성 후  
문자 API를 통해 전달



# WEB 기능 구현

## 결제 비활성화 및 실시간 주문 내역 확인

### 전체 주문 내역

주문자 : 2 / 1

주문자	주문내용	가격
김민수	이태원부대국밥	8500원

결제하기

아직 주문자가 남아있습니다.

### 전체 주문 내역

주문자 : 2 / 2

주문자	주문내용	가격
서주연	돼지목살김치찌개	9000원
김민수	이태원부대국밥	8500원

결제하기

## JavaScript / AJAX / JDBC

- 초대한 사람들이 모든 주문을 마칠때까지 결제는 비활성화 되어있는 기능
- 초대된 사람들 중 메뉴를 선택하고 전송하면 주문자의 결제리스트에 실시간으로 업데이트 AJAX로 서버와 실시간으로 연동
- 주문자를 포함한 총 인원과 주문을 마친 사람 수가 동일해야 결제하기 버튼이 활성화



# WEB 기능 구현

## 개별 포인트 차감

김민수 님

POINT : 93000

주문자 : 93000

서주연 님

POINT : 173000

초대자 : 173000

전체 주문 내역		
주문자 : 2 / 2		
주문자	주문내용	가격
서주연	해물야채튀김우동	7500원
김민수	돼지목살김치찜	9000원
결제하기		

김민수 님

POINT : 84000

주문자 : 84000

서주연 님

POINT : 165500

초대자 : 165500

## JDBC / SQL

- 리더가 최종결제버튼을 클릭하면  
DB에 저장된 주문목록에 있는 유저들이  
선택한 메뉴의 가격만큼 개인의 포인트가  
차감

--> DAO에 기재된 여러 SQL문들로 실행



# 딥러닝 모델을 활용한 기술

**SCHOOL FOOD**  
SMINGM

*About*  
**스쿨푸드**

영업시간 10:30 - 21:00  
전화번호 050712980626  
주소 서울 관악구 봉천동 1664-3

	맛	양	배달 속도	댓글 수
<b>긍정</b>	<b>698</b>	<b>795</b>	<b>482</b>	<b>824</b>
<b>부정</b>	<b>109</b>	<b>89</b>	<b>327</b>	<b>95%</b>

**친구 목록**

- ☐ 이름 : 김민수 ID : rlaalstn2 전화번호 : 01020597084
- ☐ 이름 : 김승하 ID : rlatmdyk 전화번호 : 01023998278
- ☐ 이름 : 김예슬 ID : rladptmf 전화번호 : 01020597084
- ☐ 이름 : 류윤정 ID : flblbwjd

**주문 공유**

**친구 찾기**

**주문 목록**

합계 : 0원

**주문하기**

**장조림 버터비빔밥** 가격 : 9000원  
간장맛  
Meat, Potatoes, Rice, Tomatoe  
**Order now**

맛 - 평가  
양 - 평가

**장조림 버터비빔밥** 가격 : 9000원  
매콤맛  
Meat, Potatoes, Rice, Tomatoe  
**Order now**

맛 - 평가  
양 - 평가



# 댓글 수집 & 전처리

	taste	amount	delivery	order	order_date	customer_review	owner_reivew	owner_date
0	1	1	1	열라파닭/1(뼈 or 순살 선택(순살))	5일 전	알면서도 또 시켜버렸네요 24시간집이 여기뿐이라 ... 방금 전화한 집인데 예정시간...	0	0
1	4	5	5	세트 ( 후라이드 + 겨자파닭 + 콜라 1.25L ) /1(뼈 or 순살 선택(뼈))	일주일 전	양은 확실히 많습니다 맛은 기름진부분이 있어서인지 양념이 많이 되었네요 24시간이라...	0	0
2	2	3	4	나홀로순살치킨 ( 3가지소스 + 양파슬라이스 ) /1,콜라 1.25L/1	2020년 9월 6일	순살은 다리살 같은데... 질기고... 뽀뽀하고... 여긴 좀...;	0	0
...	...	...	...	...	...	...	...	...
1505639	5	5	5	튀김족발세트 ( 튀김족발 + 막국수 + 족발볶음밥 + 보쌈무김치 + 쌈무 + 마늘 + 고추 + 쌈장 + 와사비믹스 + ...	2020년 3월 22일	이건 강추~ 다만 양이 넘 많아서 3병이서 먹었는데도 많이 남았습니다 ~ 막국수가 ...	안녕하세요 ji**님. 정말 괜찮죠 ~? 많은 고객님께서 맛있다고 하시는데 지난번 ...	2020년 3월 23일
1505640	1	1	1	튀김족발세트 ( 튀김족발 + 막국수 + 족발볶음밥 + 보쌈무김치 + 쌈무 + 마늘 + 고추 + 쌈장 + 와사비믹스 + ...	2020년 2월 10일	막국수는 넘달고 족발은 생각보다 느끼함 비추	안녕하세요 dk**님. 많은 고객님께서 "강추"라고 하시면서 좋아시는데 dk**님께...	2020년 2월 11일
1505641	5	5	5	희귀한 튀김족발 앞다리/1(음료 선택(콜라 1.25L))	2020년 2월 8일	ggggggggggggg	안녕하세요 ce**님. ㅎㅎㅎㅎㅎㅎㅎㅎ 소중한 리뷰 감사드리며 더욱 좋은 모습...	2020년 2월 11일

1505642 rows × 8 columns



# 댓글 수집 & 전처리

# 기간 : 6일

# 식당 : 1934개

# 리뷰 : 1,505,642 개

# 크기 : 394MB

# 속성 : 맛, 양, 배달 평점, 주문 음식, 리뷰,  
리뷰 날짜, 사장님 댓글, 사장님 댓글 날짜

	taste	amount	delivery	order	order_date	customer_review	owner_reivew	owner_date
0	1	1	1	열라파닭/1(빠 or 순살 선택(순살))	5일 전	알면서도 또 시켜버렸네요 24시간집이 여기뿐이라 ... 방금 전화한 집인데 예정시간...	0	0
1	4	5	5	세트 (후라이드+겨울이 닭+클라 1.25L ) /1(빠 or 순살 선택(빠))	일주일 전	양은 확실히 많습니다 맛은 기름진부분이 있어서인지 양념이 많이 되었네요 24시간이라...	0	0
2	2	3	4	나홀로순살치킨 ( 3가지소스 + 양파슬라이스 + 콜라 ) /1(빠 or 순살 선택(빠))	2020년 9월 6일	순살은 다리살 같은데... 질기고... 뽀뽀하고... 여긴 좀...;	0	0
...	...	...	...	...	...	...	...	...
1505639	5	5	5	튀김족발세트 ( 튀김족발 + 막국수 + 족발볶음밥 + 보쌈무김치 + 마늘 + 고추장 ) /1(빠 or 순살 선택(빠))	2020년 3월 22일	이건 강추~ 다만 양이 넘 많아서 3병이서 먹었는데도 많이 남았습니다 ~ 막국수가 ...	안녕하세요 ji**님. 정말 괜찮죠 ?? 많은 고객님께서 맛있다고 하시는데 지난번 ...	2020년 3월 23일
1505640	1	1	1	튀김족발세트 ( 튀김족발 + 막국수 + 족발볶음밥 + 보쌈무김치 + 마늘 + 고추장 ) /1(빠 or 순살 선택(빠))	2020년 2월 10일	막국수는 넘달고 족발은 생각보다 느끼함 비추	안녕하세요 dk**님. 많은 고객님께서 "강추"라고 하시면서 좋아하시는데 dk**님께...	2020년 2월 11일
1505641	5	5	5	희귀한 튀김족발 안다리(음료 선택(콜라 or 사이다)) /1.25L	2020년 2월 11일	안녕하세요 ce**님. ㅎㅎㅎㅎㅎ ㅎㅎ 소중한 리뷰 감사드립니다 더욱 좋은 모습...		2020년 2월 11일

1505642 rows × 8 columns



# 댓글 수집 & 전처리

	taste	amount	delivery	order	order_date	customer_review	owner_reivew	owner_date
0	1	1	1	#	문자와 공백 외 모두 제거	알면서도 또 시켜버렸네요 24시 여기까지 기다려주세요 방금 전화 한 집인데 예정시간...	0	0
1	4	5	5	#	소비자 리뷰 없는 행 제거	양은 확실히 많습니다 맛은 기름 진부분이 있어서인지 양념이 많 되... 시간이라...	0	0
2	2	3	4	#	중복 리뷰 제거	나홀로순살치킨 ( 3가지소스 + 양파슬 라이스 ) / 1, 콜라 1.25L/1 순살은 다리살 같은데... 질기 고... 뽀뽀하고... 여긴 좀...;	0	0
...	...	...	...	#	모든 영문자를 소문자로 변환	이건 강추~ 다만 양이 넘 많아서 3병이서 먹었는데도 많이 남았 습니다 ~ 막국수가 ...	...	...
1505639	5	5	5	#	모든 영문자를 소문자로 변환	튀김족발세트 ( 튀김족발 + 떡볶이 + 족 발볶음밥 + 보쌈무김치 + 쌈무 + 마늘 + 고추 + 쌈장 + 와사비믹스 + ...	안녕하세요 ji**님. 정말 괜찮조 ~~? 많은 고객님의게서 맛있다 고 하시는데 지난번 ...	2020년 3월 23일
1505640	1	1	1	#	모든 영문자를 소문자로 변환	튀김족발세트 ( 튀김족발 + 떡볶이 + 족 발볶음밥 + 보쌈무김치 + 쌈무 + 마늘 + 고추 + 쌈장 + 와사비믹스 + ...	안녕하세요 dk**님. 많은 고객 님께서 "강추"라고 하시면서 좋아시는데 dk**님께...	2020년 2월 11일
1505641	5	5	5	#	모든 영문자를 소문자로 변환	희귀한 튀김족발 앞다리/1(음료 선택(콜 라 1.25L))	안녕하세요 ce**님. ㅎㅎㅎ ㅎㅎㅎ 소중한 리뷰 감사드리며 더욱 좋은 모습...	2020년 2월 11일

1505642 rows × 8 columns



## 딥러닝

전처리 후 남은 리뷰

				고추 + 쌈장 + 와사비믹스 +
1505640	1	1	1	튀김족발세트 ( 튀김족발 + 막국수 + 발볶음밥 + 보쌈무김치 + 쌈무 + 마늘 고추 + 쌈장 + 와사비믹스 +
1505641	5	5	5	희귀한 튀김족발 앞다리/1(음료 선택(+ 라 1.25L

1082458 rows × 8 columns

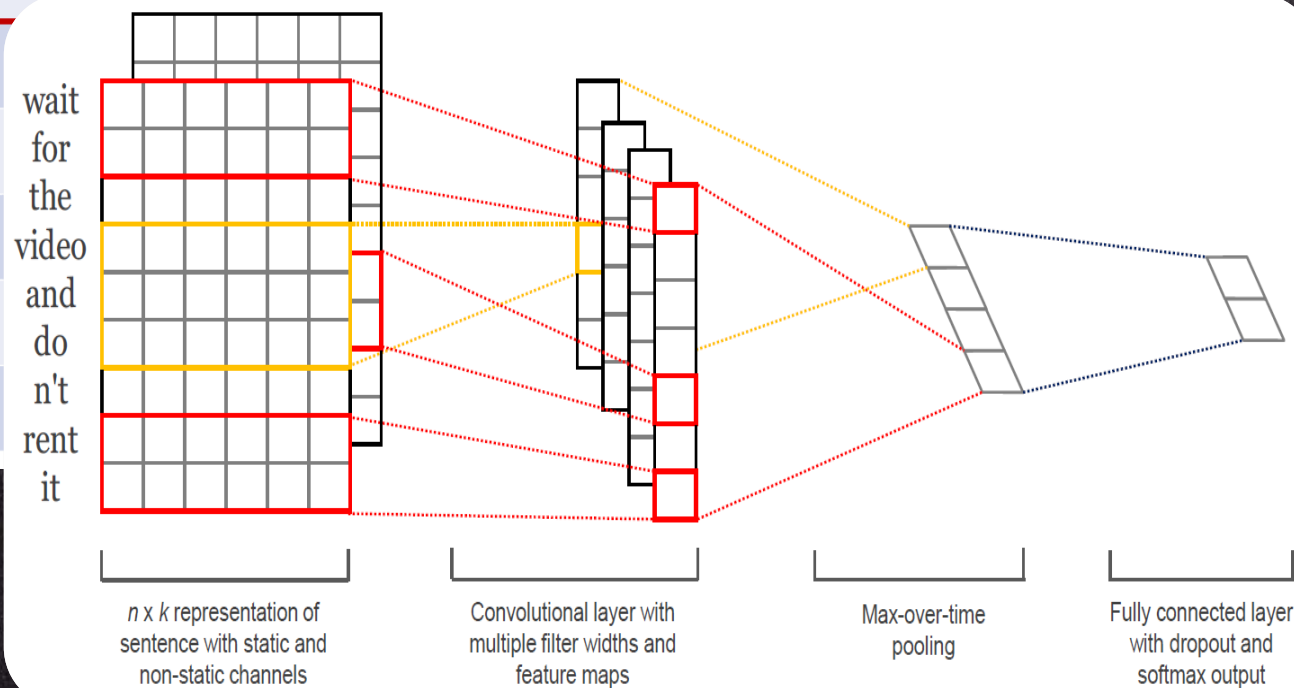
부정 댓글 수 4% 추가하기



# 딥러닝

## Multi-Kernel 1D CNN

단어	V1	V2	V3	V4	...	Vp-2	Vp-1	Vp
W1								
W2								
W3								
...								
Wn-2								
Wn-1								
Wn								





# 딥러닝

## 특성 추출

Konlpy Okt 형태소 추출기  
stem = True 정규화 수행

```
stopwords = ['의', '가', '이', '은', '를', '는', '중', '잘', '강', '과', '도', '를', '으로', '자', '에', '와', '한', '하다']

(253) okt = Okt()

X_train = []
for sentence in df[0:70000]['customer_review']:
    temp_x = []
    temp_x = okt.morphs(sentence, stem=True) # 토큰화
    temp_x = [word for word in temp_x if not word in stopwords] # 불용어 제거
    X_train.append(temp_x)

X_test = []
for sentence in df[70000:97589]['customer_review']:
    temp_x = []
    temp_x = okt.morphs(sentence, stem=True) # 토큰화
    temp_x = [word for word in temp_x if not word in stopwords] # 불용어 제거
    X_test.append(temp_x)
```

'리뷰', '민다', '시키다', '좋다', '이유', '있다',  
'^^', '앞', '단골', '되다', '것', '같다', '!', 'ㅎ'

## 정수 인코딩

Keras Tokenizer

```
print('단어 집합(vocabulary)의 크기 :', total_cnt)
print('등장 빈도가 %s번 이하인 희귀 단어의 수: %s'%(threshold - 1, rare_cnt))
print("단어 집합에서 희귀 단어의 비율:", (rare_cnt / total_cnt)*100))
print("전체 등장 빈도에서 희귀 단어 등장 빈도 비율:", (rare_freq / total_freq)*100))
```

단어 집합(vocabulary)의 크기 : 26884  
등장 빈도가 2번 이하인 희귀 단어의 수: 15992  
단어 집합에서 희귀 단어의 비율: 59.48519565540842  
전체 등장 빈도에서 희귀 단어 등장 빈도 비율: 1.1371975997272254

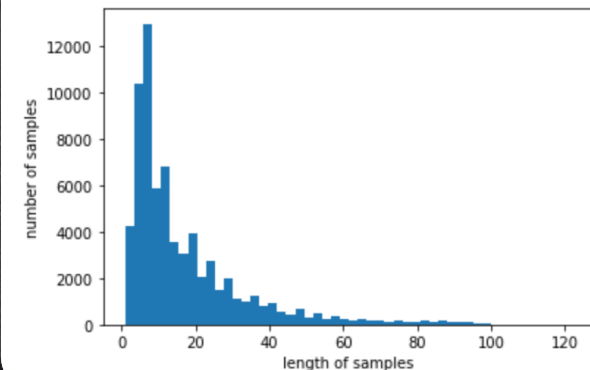
```
# 전체 단어 개수 중 빈도수 2이하인 단어 개수는 제거
# 0번 패딩 토큰과 1번 00V 토큰을 고려하여 +2
vocab_size = total_cnt - rare_cnt + 2
print('단어 집합의 크기 :', vocab_size)
```

단어 집합의 크기 : 10894

## 패딩

길이가 60 이하인 샘플의  
비율 :  
96.52684515815642

리뷰의 최대 길이 : 122  
리뷰의 평균 길이 : 17.185901648712747





# 딥러닝

▶ # Multi-Kernel 1D CNN

```
model = Sequential()
```

```
embedding_dim = 256      #1차:128
dropout_prob = (0.3, 0.4) #1차:(0.3,0.8)
num_filters = 256        #1차:128
```

```
model_input = Input(shape = (max_len,))
z = Embedding(vocab_size, embedding_dim, input_length = max_len, name="embedding")(model_input)
z = Dropout(dropout_prob[0])(z)
conv_blocks = []
```

```
for sz in [3, 4, 5]:
    conv = Conv1D(filters = num_filters,
                  kernel_size = sz,
                  padding = "valid",
                  activation = "relu",
                  strides = 1)(z)
    conv = GlobalMaxPooling1D()(conv)
    conv = Flatten()(conv)
    conv_blocks.append(conv)
```

```
z = Concatenate()(conv_blocks) if len(conv_blocks) > 1 else conv_blocks[0]
z = Dropout(dropout_prob[1])(z)
```

```
# z = Dense(1024, activation="relu")(z) 시도 1, 계속 떨어짐
z = Dense(2048, activation="relu")(z)
model_output = Dense(1, activation="sigmoid")(z)
```

```
model = Model(model_input, model_output)
model.compile(loss="binary_crossentropy", optimizer="SGD", metrics=["acc"])
```

```
es = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=30)
mc = ModelCheckpoint('best_model.h5', monitor='val_acc', mode='max', verbose=1, save_best_only=True)
```

```
history = model.fit(X_train, y_train, batch_size=128, epochs=300, validation_data = (X_test, y_test), verbose=2, callbacks=[es, mc])
```

하이퍼 파라미터  
드랍아웃

3,4,5 의 크기의 커널 256개  
Maxpooling

1 <sub>x1</sub>	1 <sub>x0</sub>	1 <sub>x1</sub>	0	0
0 <sub>x0</sub>	1 <sub>x1</sub>	1 <sub>x0</sub>	1	0
0 <sub>x1</sub>	0 <sub>x0</sub>	1 <sub>x1</sub>	1	1
0	0	1	1	0
0	1	1	0	0

Image

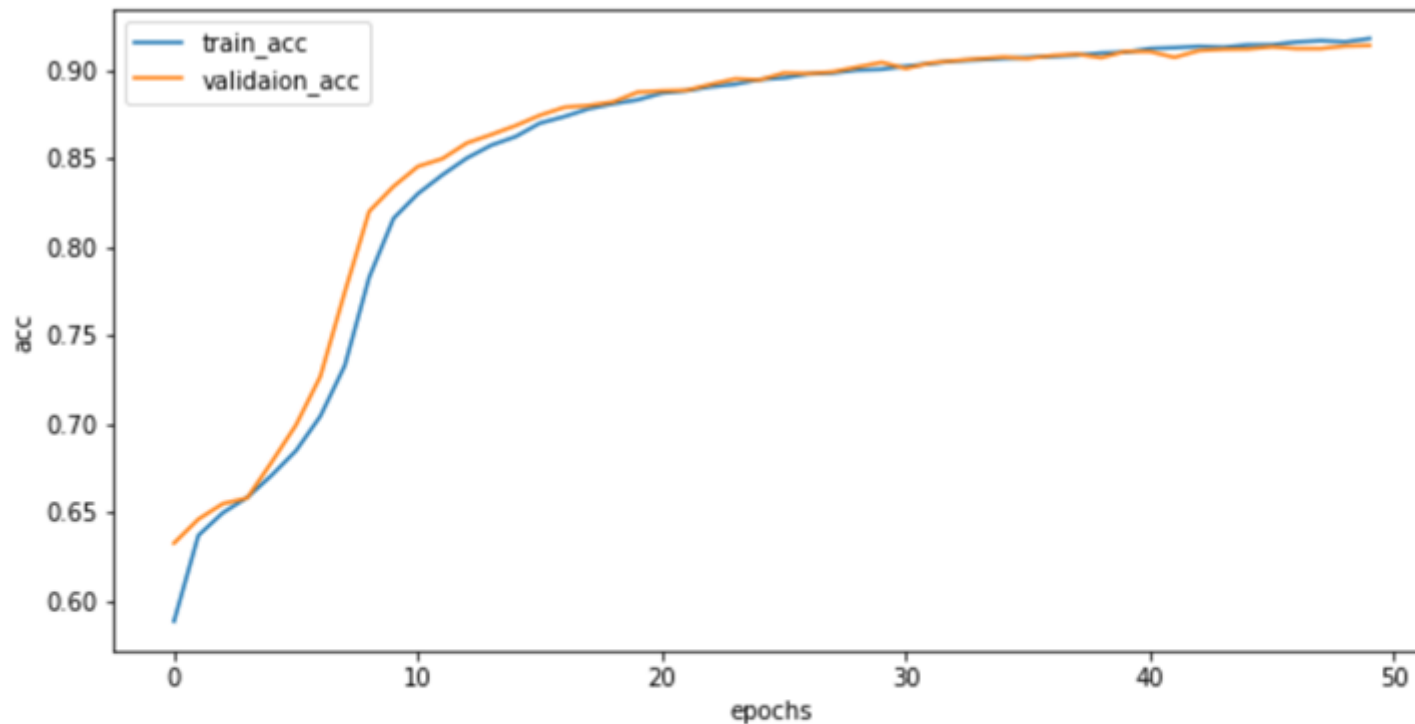
4		

Convolved  
Feature



# 딥 러닝 평가

<matplotlib.legend.Legend at 0x7fa647053e48>



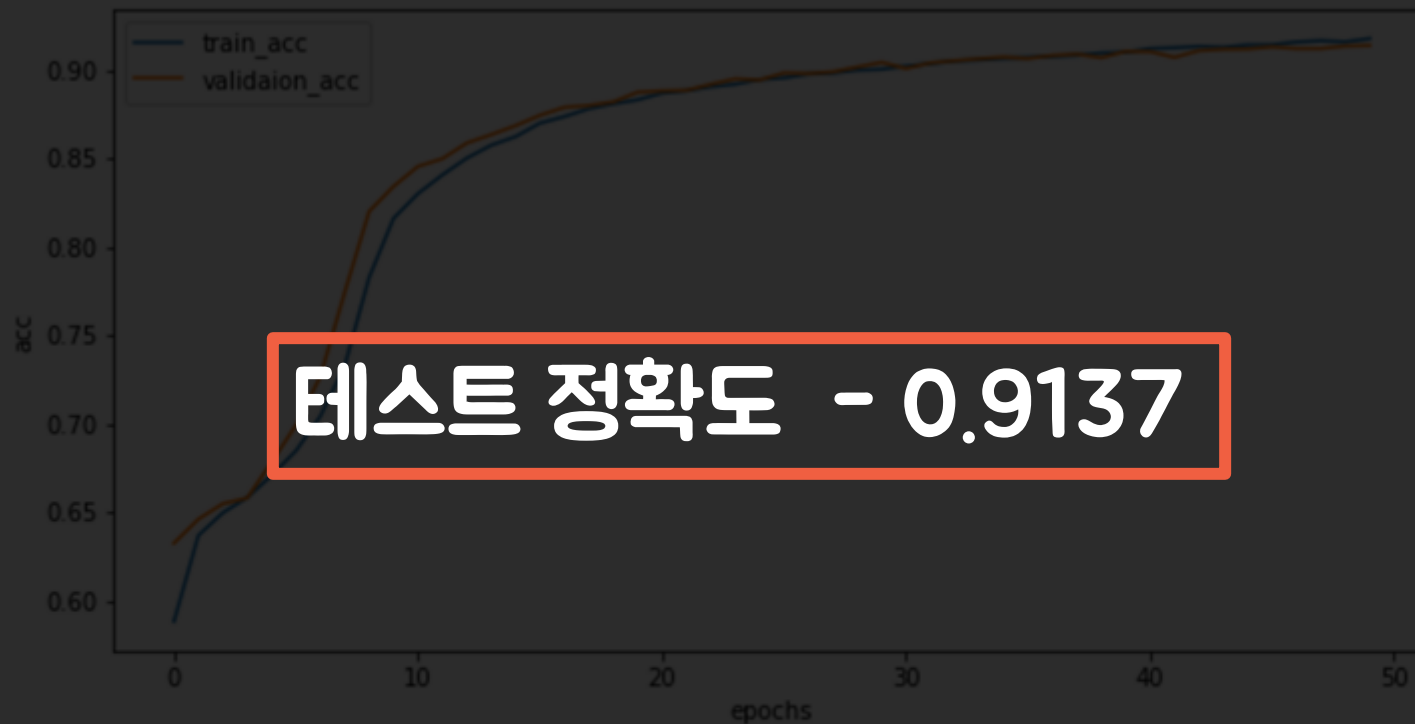
```
loaded_model = load_model('best_model.h5')  
print("테스트 정확도: %.4f" % (loaded_model.evaluate(X_test, y_test)[1]))
```

테스트 정확도: 0.9137



# 딥 러닝 평가

<matplotlib.legend.Legend at 0x7fa647053e48>



테스트 정확도 - 0.9137

```
loaded_model = load_model('best_model.h5')  
print("\n 테스트 정확도: %.4f" % (loaded_model.evaluate(X_test, y_test)[1]))
```

테스트 정확도: 0.9137



# 학습된 모델 테스트 결과

맛 평가 [ ] sentiment\_predict('맛은 좋습니다. 그렇지만 양이 정말... 굵이 비교 하자면 노랑 통닭 순살치킨 한마리 양보다 적습니다 25000원 치고는 양이 정말 참절 이네요')

75.78% 확률로 긍정 리뷰입니다.

양 평가 [31] sentiment\_predict('맛은 좋습니다. 그렇지만 양이 정말... 굵이 비교 하자면 노랑 통닭 순살치킨 한마리 양보다 적습니다 25000원 치고는 양이 정말 참절 이네요')

99.86% 확률로 부정 리뷰입니다.

맛 평가 [ ] sentiment\_predict('맛은 괜찮은듯하고 양이 적어보이지만 막상 먹으면 많아요')

90.17% 확률로 긍정 리뷰입니다.

양 평가 [ ] sentiment\_predict('맛은 괜찮은듯하고 양이 적어보이지만 막상 먹으면 많아요')

72.24% 확률로 긍정 리뷰입니다.

맛 평가 [ ] sentiment\_predict('치킨이 맛은 진짜좋아요 이렇게 맛있는치킨은 처음먹어봤습니다 근데 다들리뷰에서 말씀하시듯이 양이 적어도 너무적네요 양만 넉넉하면 좋을것같아요 잘먹었습니다')

96.77% 확률로 긍정 리뷰입니다.

양 평가 [35] sentiment\_predict('치킨이 맛은 진짜좋아요 이렇게 맛있는치킨은 처음먹어봤습니다 근데 다들리뷰에서 말씀하시듯이 양이 적어도 너무적네요 양만 넉넉하면 좋을것같아요 잘먹었습니다')

89.64% 확률로 부정 리뷰입니다.



# 머신러닝

## 특성 추출

```
[ ] from sklearn.feature_extraction.text import TfidfVectorizer
    from konlpy.tag import Okt

    okt = Okt()

    tfv = TfidfVectorizer(tokenizer=okt.morphs, ngram_range=(1,4), min_df=3, max_df=0.9)
    tfv.fit(X_train)
    tfv_X_train = tfv.transform(X_train)
    tfv_X_train
```

## LogisticRegression, GridSearchCV

```
[ ] from sklearn.linear_model import LogisticRegression
    from sklearn.model_selection import GridSearchCV

    clf = LogisticRegression(random_state = 0)

    params = {'C': [0.001, 0.01, 0.1, 1, 10, 100]}

    grid_cv = GridSearchCV(clf, param_grid = params, cv = 10, scoring = 'accuracy', verbose = 1)
    grid_cv.fit(tfv_X_train, y_train)
```



# 머신러닝

## 특성 추출

```
[ ] from sklearn.feature_extraction.text import TfidfVectorizer
    from konlpy.tag import Okt

    okt = Okt()

    tfv = TfidfVectorizer(tokenizer=okt.morphs, ngram_range=(1,4), min_df=3, max_df=0.9)
    tfv.fit(X_train)
    tfv_X_train = tfv.transform(X_train)
    tfv_X_train
```

## LogisticRegression, GridSearchCV

```
[ ] from sklearn.linear_model import LogisticRegression
    from sklearn.model_selection import GridSearchCV

    clf = LogisticRegression(random_state = 0)

    params = {'C': [0.001, 0.01, 0.1, 1, 10, 100]}

    grid_cv = GridSearchCV(clf, param_grid = params, cv = 10, scoring = 'accuracy', verbose = 1)
    grid_cv.fit(tfv_X_train, y_train)
```



# 머신러닝

## 특성 추출

```
[ ] from sklearn.feature_extraction.text import TfidfVectorizer
    from konlpy.tag import Okt

    okt = Okt()

    tfv = TfidfVectorizer(tokenizer=okt.morphs, ngram_range=(1,4), min_df=3, max_df=0.9)
    tfv.fit(X_train)
    tfv_X_train = tfv.transform(X_train)
    tfv_X_train
```

## LogisticRegression, GridSearchCV

```
[ ] from sklearn.linear_model import LogisticRegression
    from sklearn.model_selection import GridSearchCV

    clf = LogisticRegression(random_state = 0)

    params = {'C': [0.001, 0.01, 0.1, 1, 10, 100]}

    grid_cv = GridSearchCV(clf, param_grid = params, cv = 10, scoring = 'accuracy', verbose = 1)
    grid_cv.fit(tfv_X_train, y_train)
```



```
[ ] grid_cv.best_params_
```

```
↳ {'C': 10}
```

```
[ ] grid_cv.best_score_
```

```
↳ 0.9184939301972685
```

```
[ ] tfv_X_test=tfv.transform(X_test)
    grid_cv.best_estimator_.score(tfv_X_test, y_test)
```

```
↳ 0.9201593272847975
```



# 머신러닝

## 특성 추출

```
[ ] from sklearn.feature_extraction.text import TfidfVectorizer
    from konlpy.tag import Okt

    okt = Okt()

    tfv = TfidfVectorizer(tokenizer=okt.morphs, ngram_range=(1,4), min_df=3, max_df=0.9)
    tfv.fit(X_train)
    tfv_X_train = tfv.transform(X_train)
    tfv_X_train
```

## LogisticRegression, GridSearchCV

```
[ ] from sklearn.linear_model import LogisticRegression
    from sklearn.model_selection import GridSearchCV

    clf = LogisticRegression(random_state = 0)

    params = {'C': [0.001, 0.01, 0.1, 1, 10, 100]}

    grid_cv = GridSearchCV(clf, param_grid = params, cv = 10, scoring = 'accuracy', verbose = 1)
    grid_cv.fit(tfv_X_train, y_train)
```



```
[ ] grid_cv.best_params_
```

```
➤ {'C': 10}
```

```
[ ] grid_cv.best_score_
```

```
➤ 0.9184939301972685
```

```
[ ] tfv_X_test=tfv.transform(X_test)
    grid_cv.best_estimator_.score(tfv_X_test,y_test)
```

```
➤ 0.9201593272847975
```

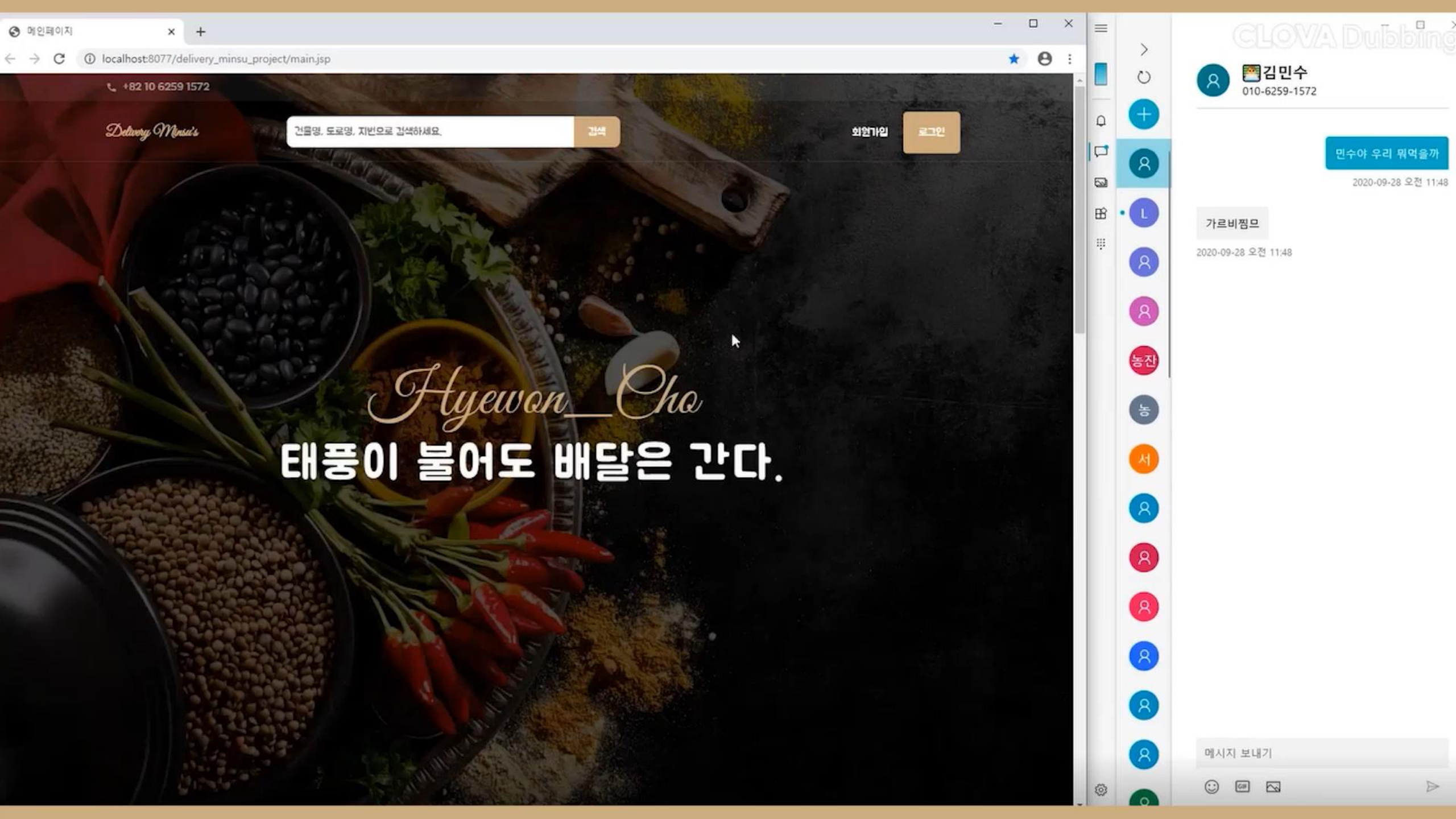


# 시연 영상

---







Hyewon Cho

태풍이 불어도 배달은 간다.



김민수  
010-6259-1572

민수야 우리 뭐먹을까

2020-09-28 오전 11:48

가르비짬뽀

2020-09-28 오전 11:48

메시지 보내기





# 기대효과 & 활용방안

---







## 기 대 효 과

---

- 영세업자의 기업체 단체 배달 활성화
- 언택트 시대 속 외식 문화가 단체 배달로 활성화
- 리뷰 데이터를 활용해 프랜차이즈 지점들의 취약점 분석
- 소비 패턴을 분석해 추후 추천 정확성 향상
- 가짜 리뷰 데이터 수집 후 선별 삭제 서비스
- 위치 기반의 식당 카테고리별 분석 서비스
- 리뷰 데이터 기반 날씨와 연동해 주문 성향 분석



# 일정 & 개발 툴

---





**추진내용**

**데이터 수집**

[illegible]



# 사용 툴





## 팀원 소개



팀장 : 최 상문

기획 총괄  
딥 러닝  
발표 및 PPT 제작



팀원 : 서 주연

DB 설계  
DAO/VO 설계  
및 JDBC  
서버 연동  
발표



팀원 : 이 한주

데이터 수집  
데이터 전처리  
딥 러닝  
PPT 제작



팀원 : 김 민수

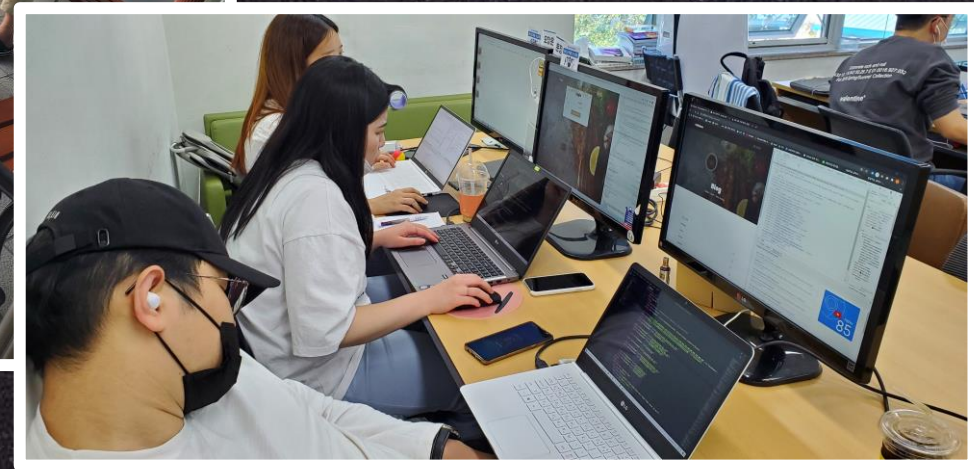
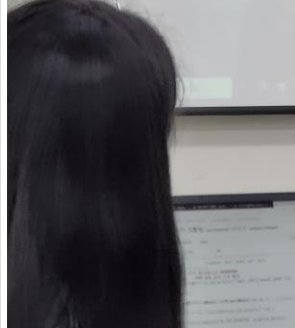
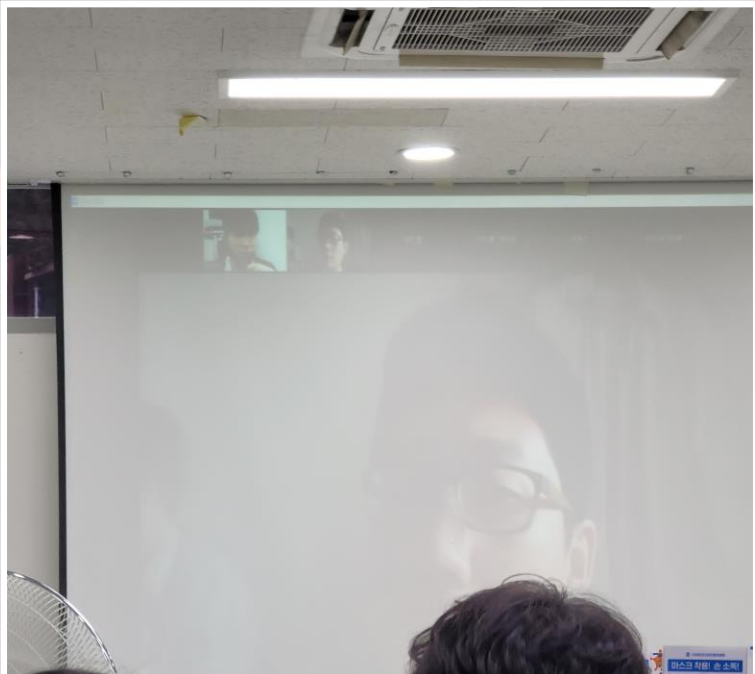
UI 설계  
웹 기능 구현 및 연동  
문자 기능 구현  
그래프 구현  
PPT 제작



팀원 : 조 혜원

UI 설계  
웹 기능 구현 및 연동  
시연 영상 제작  
PPT 제작







A top-down view of a dark, textured surface, possibly a wooden table or a stone slab, with a fine, light-colored dust or flour scattered across it. In the upper left corner, there is a small cluster of five bright red cherry tomatoes and a sprig of fresh green herbs. In the upper right corner, a portion of a pizza is visible, featuring a thick crust, melted cheese, and toppings including sliced red tomatoes, black olives, and pieces of meat. Below the pizza, a wooden spoon and a wooden fork are laid out horizontally. In the lower left corner, two triangular slices of pizza are positioned, one slightly overlapping the other, showing the same toppings as the main pizza. In the lower right corner, a small, dark-colored bowl with a gold rim is filled with a mound of white, crumbly cheese. The text "Thank you" is centered in the middle of the image, written in a clean, white, sans-serif font, and is underlined with a thin red line.

Thank you



# Q & A

---

