

2020-2 IT 집중교육2

최종 보고서

Team TomTom VIP

Recipe Helper

Team #6

소프트웨어학과 201620993 허범수

소프트웨어학과 201620486 오승민

소프트웨어학과 201620921 남도현

목차

1.	팀 및 팀원	10
1.1.	팀 명	10
1.2.	팀원 구성과 역할	10
2.	프로젝트 내용	10
2.1.	Project main idea.....	10
2.1.1.	What is the main idea of your project?	10
2.2.	Project motivations	11
2.2.1.	Why do you want to do this project?	11
2.2.2.	What is the problem with existing systems?	12
2.2.3.	Are you building a new system that does not exist now?.....	13
2.2.4.	What experience do you want to design for the target user population?	13
2.3.	Target users(stakeholders)	14
2.3.1.	Who would be your target users? List them as many as possible	14
2.3.2.	How can you access them?.....	14
2.4.	Data used	15
2.4.1.	What data are you going to use?	15
2.4.2.	How are you going to collect them?	17
2.5.	Project evaluation	18
2.6.	Project prototype	19
2.6.1.	Are you building a web or mobile system?	19
2.6.2.	How much experience does your team have in building a such system?	19
3.	Affinity Diagram	20
3.1.	Affinity Notes	20
3.2.	Grouping	20

3.3.	Group Naming	21
3.4.	Adding Experience Patterns	21
4.	Personas and Scenarios.....	23
4.1.	요리가 하고 싶은 자취생 오 나탈리씨.....	23
4.2.	맛과 영양까지 고려해 요리를 하려는 주부 오은영씨	24
4.3.	식단을 깐깐하게 지키는 직장인 김 콜먼씨	25
4.4.	대중들이 선호하는 레시피 정보를 얻고 싶은 요식업계 종사자 백중원씨	26
5.	Low-fi Prototype	27
5.1.	Sketch	27
5.2.	Low-fi Wireframe Prototype	28
5.2.1.	로그인 및 회원가입	29
5.2.2.	사용자별 레시피 추천.....	30
5.2.3.	식재료별 레시피 추천.....	31
5.2.4.	레시피 검색 & 상세보기	32
5.2.5.	스크랩 모아보기	33
5.2.6.	내 정보 확인하기.....	34
5.2.7.	게시물 등록 & 보기	36
6.	Feedback from Group 1 and complementary measure	37
6.1.	Feedback1	37
6.2.	Feedback2	37
6.3.	Feedback3	37
6.4.	Feedback4	37
6.5.	Feedback5	38
7.	Machine Learning Model Description.....	38
7.1.	Data Description (Sample, Feature).....	38
7.2.	예측하고자 하는 y 값 설명	39

7.3.	Data Preprocessing	39
7.4.	Data Labeling.....	41
7.5.	사용한 모델 알고리즘: 하이브리드 추천 시스템	42
7.6.	전체 모델 구조를 설명하는 그림 작성.....	44
7.7.	표에 관한 설명	44
7.8.	성능결과	46
7.9.	한계점 및 발전방안.....	49
7.10.	결론	60
8.	Implementation Result.....	61
8.1.	회원가입	61
8.2.	홈	62
8.3.	레시피 페이지.....	63
8.4.	나의 냉장고	64
8.5.	나의 요리책	65
8.6.	자랑하기	66
8.7.	마이페이지.....	67
9.	User Study.....	68
9.1.	목적	68
9.2.	데이터 수집 및 분석 계획	68
9.2.1.	두 가지 서비스의 비교	68
9.2.2.	사용자 스터디 설계.....	69
9.2.3.	사용자 스터디 문항	70
9.2.4.	검정 방법	71
9.3.	분석 결과	72
9.3.1.	각 서비스에 대한 만족도.....	72
9.3.2.	정규성 검정 결과(Shapiro-Wilk)	73

9.3.3.	등분산성 검정 결과(Levene).....	74
9.3.4.	검정 결과.....	74
9.4.	So what Questions에 대한 응답	76
9.4.1.	결론	82
10.	Issue and Solution.....	84
10.1.	추천 알고리즘 데이터셋 구축	84
10.2.	무료 서버	84
10.3.	한국어의 불규칙성과 불완전성	84
10.4.	레시피 정보의 부정확함.....	85
10.5.	기존 Euclidean-d의 문제	86
10.6.	레시피에 대한 영양 정보	87
11.	Reflection.....	88
11.1.	아쉬웠던 점	88
11.2.	느낀 점	88

그림 목차

[그림 1]	본 프로젝트에서 구현할 Chrome Extension의 Prototype	18
[그림 2]	Affinity notes.....	20
[그림 3]	Grouping	20
[그림 4]	Group naming	21
[그림 5]	최종 Affinity Diagram	21
[그림 6]	Persona – 오 나탈리	23
[그림 7]	Persona – 오은영	24
[그림 8]	Persona – 김 콜먼	25

[그림 9] Persona – 백중원	26
[그림 10] 1차 Sketch	27
[그림 11] Recipe Helper low-fi prototype 1	28
[그림 12] Recipe Helper low-fi prototype 2	28
[그림 13] 로그인 및 회원가입 sketch	29
[그림 14] 로그인 및 회원가입 prototype	29
[그림 15] 사용자별 레시피 추천 sketch	30
[그림 16] 사용자별 레시피 추천 prototype	30
[그림 17] 식재료별 레시피 추천 sketch	31
[그림 18] 식재료별 레시피 추천 prototype	32
[그림 19] 레시피 검색 & 상세보기 sketch	32
[그림 20] 레시피 검색 & 상세보기 prototype	33
[그림 21] 스크랩 모아보기 sketch	33
[그림 22] 스크랩 모아보기 prototype	34
[그림 23] 내 정보 확인하기 sketch	34
[그림 24] 내 정보 확인하기 prototype	35
[그림 25] 게시물 등록 & 보기 sketch	36
[그림 26] 게시물 등록 & 보기 prototype	36
[그림 27] (좌) Feature의 사용량 순으로 정리한 모습, (우) Feature의 사용량 그래프	38
[그림 28] Data Preprocessing 예시	40
[그림 29] 레시피의 DataFrame 전처리 결과	40
[그림 30] 클래스 레이블링 종류 – (1차)	41
[그림 31] 클래스 레이블링 (2차)	41
[그림 32] 최종 클래스 레이블링 결과	42
[그림 33] Euclidean Distance	42
[그림 34] SVD	42

[그림 35] Recipe-Helper Hybrid Collaborative Filtering	44
[그림 36] 레시피 – 재료 DataFrame	44
[그림 37] 사용자 – 레시피 DataFrame.....	44
[그림 38] 사용자 – Food_group DataFrame.....	45
[그림 39] 사용자 – Latest_Clicked_Recipe_by_class.....	45
[그림 40] Content Based Filtering Result.....	46
[그림 41] 실제 사용자 – 레시피 DataFrame.....	47
[그림 42] Latent Factor Collaborative Filtering	47
[그림 43] [그림 42]의 결과를 내림차순으로 정렬한 후 상위 10개를 추출한 결과	48
[그림 44] 사용자 추천 결과.....	48
[그림 45] 추천 알고리즘의 문제	49
[그림 46] 서버 성능의 한계로 계산 범위를 줄이는 과정	50
[그림 47] 분산 시스템 예시	50
[그림 48] 레시피 정보의 부정확함	51
[그림 49] 예시: 좌측(동일한 요리의 유사도 측정 문제), 우측(유사한 요리의 유사도 측정 문제) .	51
[그림 50] 예시: 주재료의 가중치 조정.....	52
[그림 51] 예시: Food_group을 Feature로 추가하여 가중치 조정한 결과	52
[그림 52] Recipe Helper의 영양정보 획득 과정	53
[그림 53] 레시피 DB 예시.....	54
[그림 54] 레시피의 제목을 Tokenizing한 결과	55
[그림 55] 예시: Tag 추출을 위한 Tokenizing list.....	56
[그림 56] Tag 추출 결과1	56
[그림 57] Tag 추출 결과2	57
[그림 58] Tag 추출 결과 3	57
[그림 59] 예시: 음식 및 레시피에 대한 특징 추출	58
[그림 60] 성별/연령별 추천 모델	58

[그림 61] 식재료별 추천	59
[그림 62] 식재료 기반의 레시피 추천 결과	59
[그림 63] Ecommerce와의 연동으로 발전 방안	60
[그림 64] 카카오 로그인, 회원가입, 취향분석1, 취향분석2	61
[그림 65] 흠, 펼쳤을 때, 검색할 때, 검색 결과	62
[그림 66] 레시피 페이지, 요리책 추가, 영양정보 확인	63
[그림 67] 냉장고 화면, 식재료 수정 화면, 식재료 추가, 레시피 검색	64
[그림 68] 요리책 화면, 스크랩 해제, 일부 스크랩 다시 선택, 반영 결과	65
[그림 69] 글쓰기 사진2개 선택, 사진1개 삭제, 자랑하기 화면, 댓글 남기기	66
[그림 70] 마이페이지, 건강상태 및 알레르기 상태 변경, 워드클라우드 변경, 설문조사 링크	67
[그림 71] Recipe Helper main(좌), 만개의 레시피 메인화면(우)	68
[그림 72] 사용자 스터디 파악 단위별 문항	70
[그림 73] 검정 절차	71
[그림 74] 만족도 통계 – Recipe Helper	72
[그림 75] 만족도 통계 – 만개의 레시피	72
[그림 76] Box Plot 결과 (좌) Recipe Helper (우) 만개의 레시피	73
[그림 77] 정규성 검정을 위한 코드와 결과	73
[그림 78] 등분산성 검정을 위한 코드와 결과	74
[그림 79] Independent T test, Welch T test 코드	74
[그림 80] Wilcoxon rank-sum test 코드	75
[그림 81] 정규성 검사, 등분산성 검사, T 검정 자동화 코드	75
[그림 82] Q6 분석 결과	76
[그림 83] Q6 설문 결과 (좌) 만개의 레시피 (우) Recipe Helper	76
[그림 84] Q7 분석 결과	77
[그림 85] Q7 설문 결과 (좌) 만개의 레시피 (우) Recipe Helper	77
[그림 86] Q9 분석 결과	77

[그림 87] Q9 설문 결과 (좌) 만개의 레시피 (우) Recipe Helper	78
[그림 88] Q10 분석 결과	79
[그림 89] Q10 설문 결과 (좌) 만개의 레시피 (우) Recipe Helper	79
[그림 90] Q11 분석 결과	80
[그림 91] Q11 설문 결과 (좌) 만개의 레시피 (우) Recipe Helper	80
[그림 92] Q13 결과 분석	81
[그림 93] Q13 설문 결과 (좌) 만개의 레시피 (우) Recipe Helper	81
[그림 94] Q14 결과 분석	82
[그림 95] Q14 설문 결과 (좌) 만개의 레시피 (우) Recipe Helper	82
[그림 96] 문항3 설문결과(좌), 문항4 설문결과(우).....	82
[그림 97] 유사한 레시피의 식재료	85

표 목차

[표 1] 기능 별 input data와 output data	16
[표 2] Recipe_ID – User_ID click 표	39
[표 3] Recipe ID별 유사도	39
[표 4] 첫 번째 문제의 예시.....	86
[표 5] 두 번째 문제의 예시.....	86
[표 6] 가중치를 조정한 예시	87

1. 팀 및 팀원

이 장에서는 팀 이름과 팀원 구성과 역할에 대해 간략히 설명한다.

1.1. 팀 명

학교 앞에 유일하게 있는 24시간 카페가 TomTom이어서 매일 그곳에서 밤을 셀 줄 알고 TomTom VIP (TOM'N TOM's VIP)로 지었다.

1.2. 팀원 구성과 역할

팀원은 오승민(조장), 허범수, 남도현으로 이루어져 있으며, 기존에는 각자 역할을 나누었지만 같이 배우며 진행하자는 차원에서 안드로이드와 서버, 머신러닝 모델 개발, UI/UX까지 전체 개발 과정을 세 명이 함께 하였다.

2. 프로젝트 내용

먼저 프로젝트의 핵심이 되는 아이디어와 프로젝트를 계획하게 된 동기에 대해 설명한다. 동기에는 이 프로젝트를 하고 싶은 이유와 기존 시스템의 존재 유무, 기존 시스템이 가지고 있는 문제 등이 포함된다. 그 후 목표로 하는 사용 계층을 설정하고, 실제로 시스템 구축을 위해서 우리 팀원들이 가지고 있는 역량과 필요한 개발 내용을 설명한다.

2.1. Project main idea

2.1.1. What is the main idea of your project?

'먹는 행위'는 인간의 본능이다. 인간에게 음식은 가장 원초적인 관심사다. 하지만 '먹는 행위'에 대한 가치 · 철학은 시대에 따라 달랐다. 생활 수준이 대체로 낮았던 과거에는 생존과 일을 하기 위해 음식의 효율이 중요했다면, 과학기술의 발전으로 급속한 경제성장을 이루한 결과로 풍족한 삶을 누리게 된 인류는 '삶의 질'의 관점에서 음식을 다시 바라보게 되었다.

스웨덴의 가구 제조 기업인 '이케아'는 완성된 가구 제품이 아닌 조립하는 가구를 판매한다. 가구를 손수 조립해야 한다는 사실은 소비자 입장에서 불편하고 귀찮은 일이다. 그런데 이 불편함이 오히려 제품에 대한 충성심과 애정을 높인다. 귀찮은 조립을 성공적으로 마무리했을 때 소비자는 기성 제품을 샀을 때와 비교되지 않은 뿌듯함과 애정을 느낀다. 사람들은 완성된 제품의 편

리성과 기능성보다 그것에 담긴 노력과 정성을 더 가치 있게 평가한다는 것이다.

직접 요리를 만들어 먹고 싶어하는 심리는 이런 '이케아 효과'의 좋은 예다. 산골과 바닷가에서 요리 재료를 직접 키우거나 채취해 음식을 만드는 과정은 마치 조립 가구를 완성하는 과정에서 맛보는 희열을 선사한다. 완성된 요리의 맛뿐만 아니라 요리를 만드는 과정의 노력과 정성이 시청자들의 마음을 사로잡는다.

지난해 1인 가구가 600만 명을 돌파하면서 직접 요리하는 사람이 늘어나기 시작했고 유명한 요식업 사업가가 TV와 YouTube에 나오면서 요리에 대한 대중의 관심이 뜨거워지기 시작했다. 이런 상황 속에서 코로나 19로 사람들이 외식하는 비중이 줄고 개인의 여가활동이 제약되는 상황에서 취미 활동으로서 요리가 주목받고 있다. 특히 요리에는 정답이 없고 똑같은 요리를 하더라도 무궁무진한 방법이 존재하며 경험과 문화에 따라 '개성'이 묻어나기 때문에 요리는 새로운 지식과 가치의 창출 및 공유를 촉진시킨다는 점에서 그 의미가 크다.

이러한 생각에 근거하여 본 프로젝트에서는 요리를 즐겨하고 건강하게 식단을 구성하는 사람들을 위해 개개인의 취향과 상황 예측할 수 있는 정보를 이용하여(남은 식재료, 관심, 건강 상태, 연령 등) 사용자에게 맞춤형 레시피를 추천하도록 하는 요리 레시피 공유 서비스를 시도해보고자 한다. 추가로 개인은 자유롭게 음식 레시피를 사진이나 동영상을 함께 첨부하여 작성하고, 작성된 레시피는 회원들 사이에서 공유되고, 사람들이 직접 레시피를 사용하여 요리를 시도한 후 레시피에 대해 후기를 남기는 등 평가할 수 있다.

2.2. Project motivations

2.2.1. Why do you want to do this project?

Z세대가 본격적 사회 활동을 시작하는 시대로 접어들었다. 일반적으로 Z세대라 불리는 90년대 중후반부터 2006년까지 탄생한 세대는 자신의 개성을 표출하기를 좋아하고, 관심사나 소비활동, 가치관 등 자신의 성향을 사람들과 공유하고 싶어한다. 또한 1인가구 증가로 혼자 요리를 해 먹는 사람의 비율이 늘어나고, 그에 따라 혼자서도 맛있게 한 끼를 해결하고 싶어하는 사람들의 욕구 역시 증가했다. 이를 잘 보여주는 것이 수년 전, 그리고 현재까지도 진행중인 백종원 열풍이다. 백종원이 출연하는 프로그램마다 성공을 거두고, 그가 운영하는 자신의 유튜브 채널 역시 450만 명 이상의 구독자를 보유하고 있다. 그가 유튜브에서 소개하는, 혼자 따라하기 쉽고 쉽게 구할 수 있는 재료로 만드는 자취생 요리는 올리는 족족 '백종원 레시피'라는 이름으로 수많은 블로그와 티스토리에 소개된다.

작년 말부터 시작된 코로나 사태는 사람들의 활동 반경을 오프라인에서 온라인으로 이동하게 했다. 외식활동이 줄어들어서 밖에서 저녁을 사 먹던 사람들도 집 앞 부엌에서 자신이 요리를 해야 하는 상황이 온 것이다. 하지만 앞서 말한 '백종원 레시피'만 따라하다 보면 냉장고에 쌓여가는 재료는 점점 많아질 것이다. 어느 순간에는 남아있는 재료들만으로 요리를 하는 상황이 올 때,

사람들은 적절한 요리를 떠올리기 쉽지 않다. 결국 매번 볶음밥으로 마무리하는 사람들이 많았다.

이러한 문제는 특히 1인가구에서 많이 일어나는 일 이기도 하다. 보통 자취생의 경우 식자재를 구매해 요리를 해도 1인분의 양은 많지 않아 남기 일쑤이다. 먹고 싶은 요리를 계속 하다 보면 냉장고에는 남아있는 재료로 꽉 차서 처치 곤란이다. 이 상황에서 남는 재료를 가지고 맛있는 요리를 해먹기에는 요리 전문가가 아닌 이상 떠올리기 쉽지 않다.

우리는 이 현상에 집중해서, 현재 냉장고에 있는 식재료와 그 양을 고려해서 다양한 레시피의 조합을 추천해 주는 시스템이 있으면 좋을 것이라고 생각했다. 또 이 시스템의 예상 사용자는 요리에 관심이 많고, 냉장고에 음식 재고가 쌓이는 게 싫은 사람으로 정했다. 또한 정확한 영양 정보가 나와있지 않은 레시피들이 많아 사용자에게 레시피의 재료를 통해 정확한 영양 정보를 제공해주고 싶었다. 이는 평소에 건강하게 식사를 하고 싶어하는 사람들까지 타겟층을 넓힐 수 있다. 한편, 확장성을 고려하여 본 프로그램을 사용하는 사용자가 많아진다면 사용자들끼리 자신이 스스로 연구해서 만든 레시피나, 어떤 요리를 더 맛있게 할 수 있는 팁을 다른 사람과 공유할 수 있는 기회가 주어진다면 좋을 것이다.

식품의 영양 성분을 계산하거나, 사용자 맞춤화 요리 레시피를 추천해 줄 때 빅데이터와 인공지능 기술을 적용할 예정이다. 식품의 영양 성분을 계산해주고, 사용자의 건강 상태에 따라 적절한 영양 성분으로 구성된 레시피를 추천한다. 또한, 자신이 구매해 온 식재료의 이름과 양을 처음에 입력하면, 레시피 하나를 조리할 때 마다 남는 양을 계산해서 남은 재료를 고려해 이 재료들로 만들 수 있는 또 다른 레시피를 추천해 준다. 마지막으로 추천 이후 추천해 준 레시피가 만족스러운지 설문할 수 있는 팝업을 띄워서 결과를 분석해, 각 개인의 특성에 따라 맞춤형 서비스를 제공 가능하다.

2.2.2. What is the problem with existing systems?

'요리백과 만개의 레시피'

500만명 이상의 유저를 보유하고 있는, 레시피 관련 검색 결과 처음에 노출되는 레시피 제공 사이트이다. 요리 종류, 현재 상황, 건강상태, 사용자의 유형에 따른 분류 항목이 아주 상세하게 들어가 있다. 하지만, 실제로 분류된 항목들을 들여다보면 분류 항목만 많고 제대로 분류되어 있지 않은 경우가 많다. 또한 여러가지 상황을 고려하지 못하고 단 한가지의 상황에 대해서만 항목을 분류하여 레시피를 제공한다. 하나의 예를 들어보면 만개의 레시피에는 약 10만개 이상의 레시피가 존재하지만 그 중 자취생을 위한 레시피는 60개뿐이다.

이렇게, 만개의 레시피 사이트와 더불어 레시피의 종류가 1,000개를 넘어가는 사이트들은 레시피에 대해 항목별로 세세하게 분류는 되어있지만 그 안은 제대로 분류되지 않는 음식들이 많다. 또한, 항목이 매우 많다는 것은 사용자는 혼란을 느끼기 쉬우며 자신이 선호하는 요리의 레시피

를 구하기 위해 해야 할 일이 많아질 수 있다는 의미이다. 수많은 항목들을 일일이 확인하면서 요리 레시피를 검색하는 것이 쉬운 일은 아니다. 결국, 사용자는 웹 페이지의 가장 상단에 노출되어 있는 요리의 레시피를 만들 가능성이 커지고 자신의 상황을 고려하지 않은 채 요리를 만들면 재고로 쌓이는 식재료가 많아질 우려가 있다. 또한, 식재료의 영양정보를 따로 제공하지 않기 때문에, 자신의 건강 상태에 관심을 가지는 사용자에게는 아쉬울 수 있다.

이처럼, 사용자 중심의 레시피 추천을 해주는 국내 레시피 제공 사이트는 존재하지 않으며, 사용자의 남는 식재료를 고려해주는 레시피 제공 사이트에서도 찾아볼 수 없었다. 따라서, 팀-6은 사용자 중심의 레시피 추천 서비스를 제공할 예정이며, 레시피 사이트를 찾아보며 요리를 해먹는 경우가 많은 자취생(아주대학생, 약 10명)을 대상으로 간단하게 물어봤을 때 매우 긍정적인 반응을 볼 수 있었다. 이 서비스는 사용자의 건강 상태나 성별, 나이, 음식 선호도, 냉장고 속 남은 식재료 등을 통해 사용자 맞춤형 레시피를 제공해주는 서비스로, 다른 레시피 사이트에서는 볼 수 없는 차별화된 서비스이다. 뿐만 아니라, 추후 레시피를 공유할 수 있는 서비스 또한 제공하여, 다양한 요리 레시피들을 사람들이 공유하고 의견을 들을 수 있다면 지속적으로 TomTom VIP팀의 서비스를 이용하도록 사용자를 확보하고 고유 시스템으로 자리 잡을 것이다.

2.2.3. Are you building a new system that does not exist now?

만약 이 시스템의 정체성을 단순히 요리 레시피 공유나 자취생의 생활에 도움을 주는 시스템으로 정의한다면, 이런 시스템은 기존에 아주 많이 있다. 요리 레시피를 소개하는 앱은 구글 플레이에 치면 아주 많이 나오고, 레시피 공유 앱 역시 다수가 존재한다. 자취생의 생활에 이로운, 요리를 포함한 생활 정보 소개 시스템도 존재하고, 혼자 지내는 1인 가구에게 제휴 쿠폰이나 이벤트를 소개해 주는 앱도 존재한다. 그렇지만 요리 레시피 정보를 얻고, 자신의 레시피를 공유하고, 사용자의 냉장고 상황까지 생각해 주는, 자주 요리를 하거나 1인 가구인 자취생에게 특히 유용한 레시피 전문 앱이 있느냐고 한다면, 그런 시스템은 현재 조사한 바로는 존재하지 않는다. 또, 식재료의 상황과 사용자의 연령, 관심, 건강상태, 그리고 계절 등 인공지능 도입을 통해 분석 후, 맞춤형 추천을 해 주는 서비스 역시 찾아보기 힘들었다. 사용자에게 레시피를 추천한 이후, 추천된 레시피의 만족도를 likert scale로 평가하도록 해서 피드백 결과를 모델에게 학습하도록 하는, 사용자 개개인을 고려한 레시피 추천을 해 주는 서비스도 없다. 사용 계층을 아주 세분화해 문제 상황을 고민하고, 문제를 해결하는 데 머신 러닝 기술의 도입은 매우 긍정적이며 이전에는 없던 기술을 만들어낼 수 있게 한다.

2.2.4. What experience do you want to design for the target user population?

- 사용자가 냉장고에 어떠한 식재료가 있는지 시스템에 입력한다면, 식재료를 추가로 구입할 필요 없는 요리를 추천 받는다.

- 사용자가 처음 구매한 식재료를 시스템에 입력하고 레시피 사이트에서 제공하는 요리를 만들었을 시, 자동으로 식재료의 상황을 업데이트 한다. 이를 통해 자신의 냉장고 상황을 더 직관적으로 알 수 있다.
- 사용자는 자신의 냉장고 상황을 한 눈에 파악할 수 있고, 남은 재료를 쉽게 사용할 수 있어서 식재료가 상해서 버리는 일이 줄어들 것이고 결과적으로 배출되는 음식물 쓰레기의 양을 줄일 수 있다.
- 가입시 사용자를 Grouping할 수 있는 성별, 나이 등의 정보를 입력하고 자신이 선호하는 음식이 무엇인지 선택하는 기초 조사를 실시한다. 뿐만 아니라, 사용자가 레시피에 대해 조회를 했을 때 음식 및 레시피에 대해 관심정도를 설정할 수 있도록 하여 사용자가 선호하는 레시피를 정확하게 추천하는 시스템을 제공할 수 있다.
- 식재료를 적절한 시기에 사용할 수 있으므로 더 신선한 상태로 요리할 수 있고, 영양 정보가 제공되기 때문에 균형 있는 영양소 섭취를 격려할 수 있다.
- 자신이 만든 레시피 중 더 괜찮다고 생각되는 점이 있다면 다른 사용자들과 공유하여 의견을 듣고 피드백을 받으며 자신만의 요리를 발전시킬 수 있다. 자신이 관심있는 분야에서 긍정적인 경험을 하게 된다면 장기적으로 서비스를 계속 이용할 수 있다.

2.3. Target users(stakeholders)

2.3.1. Who would be your target users? List them as many as possible

- 1) 자녀를 위해 요리를 자주하는 주부
- 2) 평소 요리에 관심이 많고 취미로 즐기는 사람
- 3) 외부 활동의 제약으로 요리에 관심이 생긴 학생 및 직장인
- 4) 건강에 관심이 많고 건강하게 식사를 하고자 하는 사람
- 5) 냉장고에 남아있는 식재료를 최대한 활용하여 요리를 하고자 하는 사람
- 6) 요리에 대한 다양한 인사이트와 트렌드를 원하는 요리사 및 요식업 종사자

2.3.2. How can you access them?

코로나 사태로 인해 오프라인 조사 보다는 온라인 매체를 활용한 조사가 더욱 활발하게 진행될 것으로 예상했다. 오프라인 조사보다 참여를 요구하는 행위를 하기는 더 힘들다고 생각했지만, 더 많은 설문자에게 요구할 수 있고 참여자의 충도 더욱 넓힐 수 있어서 긍정적인 결과를 얻을 수

있을 것으로 예상한다.

- 1) 지인에게 요청하거나 지역 커뮤니티나 맘 카페 등을 통해 인터뷰와 설문조사를 시행할 수 있다.
- 2) 지인과 학교 커뮤니티, 대학생이 주로 사용하는 커뮤니티에 인터뷰와 설문조사를 시행할 수 있다.
- 3) 직접 요리사분께 찾아갈 수도 있고, 이메일을 통해 인터뷰와 설문조사를 시행할 수 있다.
- 4) 대학생이나 기숙사생 대상으로 게시판을 활용해서 조사를 할 수 있다.
- 5) 아주대학교 내에서 행인들 대상으로도 인터뷰와 설문조사를 시행할 수 있다. 그러나 코로나 사태로 인해 불특정 다수와의 접촉은 위험하므로 가장 후순위의 방법이 되어야 한다.

2.4. Data used

2.4.1. What data are you going to use?

본 프로젝트에서 제공하는 주요 기능은 크게 4가지이다.

1. 사용자의 냉장고에 남아있는 식재료를 입력하면 이를 통해 만들 수 있는 요리 추천.
2. 사용자 상태(나이, 성별, 관심 요리 히스토리 etc.)를 종합하여 자동화된 요리 추천.
3. 전날 먹은 영양 상태를 토대로 부족한 영양소를 보충하는 요리 추천.
4. 영양 정보가 제공되지 않는 요리의 영양 정보 생성 및 시각화
5. 한 레시피 사이트에서만 찾는 것이 아닌 다양한 사이트의 레시피를 검색해서 사용자에게 추천한다.

첫 번째 기능은 사용자가 자신의 냉장고에 남아 있는 식재료를 입력했을 시, 추가적으로 재료를 구매하지 않고 만들 수 있는 요리를 제공해주는 기능이다. 이 기능을 제공하기 위해서는 사용자의 냉장고 상황 data와 재료가 정확히 적혀 있는 레시피 data가 필요하다.

두 번째 기능은 사용자가 관심을 가질 만한 요리를 추천해주는 기능이다. 이 기능을 제공하기 할 때에는 사용자의 나이, 성별, 건강상태, 클릭한 요리의 히스토리, 사용자의 요리에 대한 평점 정보 등을 이용할 예정이기 때문에 사용자의 신상 정보 data와 클릭한 요리의 히스토리 data, 개별 요리에 대한 사용자의 평점 등의 정보가 필요하다.

세 번째 기능은 사용자가 전날 먹은 음식에 대해 영양 정보를 계산한 뒤, 부족한 영양 성분을 보충할 수 있는 요리의 레시피를 추천해주는 기능이다. 이 기능을 제공하기 위해서는 사용자가 전날 섭취한 데이터와 레시피 별 영양 정보가 필요하다.

네 번째 기능은 두 번째 기능에서 사용자의 건강상태에 따라 요리를 추천해주기 위해 필요한

데이터를 구축하기 위해 필요한 기능이다. 다만, 건강하게 식사를 원하거나 다이어트를 원하는 사람들 또한 요리의 영양정보를 자주 필요로 하다고 생각되어 따로 사용자에게도 제공하는 기능을 추가할 것이다. 이 기능을 제공하기 위해서는 직접 영양정보를 계산할 수 있어야 하므로 요리에 사용되는 재료 data와 재료별 영양 정보 data가 필요하다. 따라서 필요한 data는 다음과 같다.

1. 사용자 정보(나이, 성별, 키, 몸무게, 질병, 클릭 히스토리, 냉장고 상황)
2. 레시피
3. 요리 재료
4. 재료별 영양 정보
5. 레시피 별 영양 정보

아래 표는 기능별 필요한 데이터와 생성되는 데이터를 정리한 표이다.

기능	필요한 데이터	생성 데이터
식재료를 통한 요리 추천	남아있는 식재료, 레시피 재료	당장 만들 수 있는 요리 레시피
사용자에 따른 요리 추천	사용자의 정보(나이, 성별, 키, 몸무게, 질병, 클릭 히스토리, 냉장고 상황), 사용자의 음식 선호(음식에 대한 평점), 레시피 재료, 레시피 별 영양 정보	사용자에게 맞는, 사용자가 좋아할 요리 레시피
전날 섭취한 영양상태에 따른 요리 추천	사용자가 요리한 데이터 및 섭취한 요리 데이터, 레시피 별 영양 정보	오늘 먹으면 좋은 요리 레시피
영양 정보 시각화	레시피 재료, 재료별 영양 정보	요리의 영양 정보

[표 1] 기능 별 input data 와 output data

이 중 사용자에 따른 요리 추천 기능은 추천 Model을 만들어서 ML기반의 알고리즘을 구현할 것이다. 이 때 사용할 input은 사용자의 정보이고 output은 사용자 정보에 따른 추천 요리이다. Model을 트레이닝하기 위해서 사용할 input data는 레시피 재료와 사용자 정보를 사용할 것이고 output data는 요리의 레시피이다. 이때, 사용자 정보의 데이터 구성은 나이, 성별, 선호하는 음식 종류, 클릭 히스토리, 냉장고에 남아 있는 식재료로 구성된다. 선호하는 음식의 종류를 판별하는 것은 로그인 시 수행될 음식 취향 분석 데이터와 요리 레시피를 검색하면서 수행할 수 있는 선호도 조사 데이터를 통해 판별한다.

2.4.2. How are you going to collect them?

사용자의 정보는 회원 가입을 통해 사용자로부터 입력 받고 냉장고 상황은 수시로 사용자에 입력받을 수 있도록 한다. 그리고 사용자가 레시피 목록에서 한 요리를 만들었다고 생각하면 자동으로 사용자의 냉장고 상태를 업데이트 해주는 방식으로 data를 업데이트 해준다.

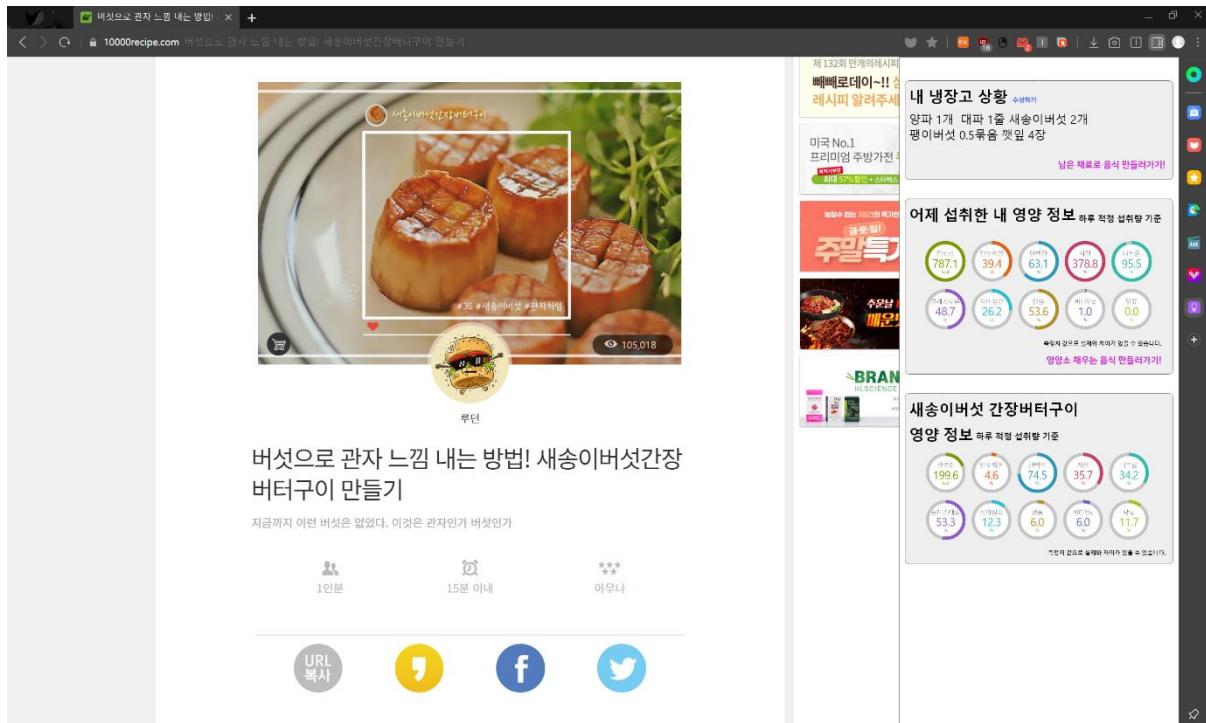
레시피와 요리의 재료는 관련 사이트에서 크롤링을 통해 구한다. 관련 사이트는 '만개의 레시피 (<https://www.10000recipe.com/index.html>)', '해먹(<https://www.haemukja.com/#>)' 등의 사이트가 있다. 현재 국내에서 레시피를 제공하는 사이트는 위 2개가 가장 활성화 되어있고 정확한 정보를 제공하여 위 사이트만을 사용한다. 해외에도 레시피를 제공하는 사이트가 존재하지만 언어적인 장벽과 음식에 사용되는 재료가 우리나라에는 없을 가능성이 높은 문제 때문에 해외 레시피 사이트는 배제했다. 또한 요리에 대한 정보가 올라오는 SNS에서도 정보를 얻을 수 있는 방법이 있지만 레시피라는 정보를 만드는 데 있어 필요한 재료, 조리과정 등과 같이 유의미한 정보를 모두 포함한 게시글이 많지 않다는 점, 그리고 SNS 특성상 유행하는 음식 위주로만 올라온다는 점에서 SNS에서 데이터를 얻지 않기로 결정했다. 그러나 이로 인해 동일한 음식에 대해 여러 개의 레시피가 존재하여 전체적인 음식의 개수가 적을 수 있는 문제가 있는데, 이는 해당 사이트에서 카테고리별로 일반적인 대중이 인식하고 있는 대부분의 음식뿐 만 아니라 이용자수가 많아 최근 대중매체를 통해 소개되어 유행하는 음식 또한 빠르게 반영되어 포함하고 있기 때문에 절대적인 음식의 개수 및 재료의 개수는 충분하다고 판단된다. 만개의 레시피 기준으로 하나의 레시피당 크롤링 했을 시 1kb정도의 데이터 용량을 가지고 있다. 총 145,194개의 레시피가 존재하여 $105194\text{kb} = 141.8\text{mb}$ 의 크기의 데이터를 확보할 수 있다.

재료별 영양 정보는 식품의약품안전처에서 xlsx 형식으로 다운 받거나 open API를 이용해 다운 받을 수 있다. xlsx 형식으로 받을 경우 47.18mb정도의 데이터를 구축할 수 있으며 이 데이터는 음식, 가공식품, 농축산물, 수산물로 구분되어 있다.

레시피별 영양 정보는 재료와 재료의 영양 정보를 토대로 본 팀에서 직접 계산하는 알고리즘을 구현할 예정이다. ML을 사용할지 공식을 이용하여 계산할 지 정해지지 않았지만 '해먹'사이트에서는 재료와 영양 정보를 같이 제공하므로 정확한 계산 알고리즘을 구현할 수 있다.

선호도 조사는 '해당 음식을 만들어 먹고 싶은가?' 혹은 '선호하는가?'등의 질문을 통해 1~5점의 점수를 매기는 방식으로 진행한다. 점수를 매길 때, 1점을 주는 항목에 대해서는 앞으로의 사용자 추천 목록에서 제외하고, 2~5점은 점수에 따라 가중치를 달리 주는 방식으로 Model의 weight을 조정한다.

음식의 종류를 나누는 방법은 레시피 DB를 구축한 후, 요리의 이름을 단어 단위로 split한 뒤 많이 나온 단어를 나열 한 후 해당 단어가 요리라고 판단되고, 모든 레시피의 종류를 분류할 수 있을 때 까지의 단어를 음식의 종류로 분류한다.



[그림 1] 본 프로젝트에서 구현할 Chrome Extension 의 Prototype

2.5. Project evaluation

기존의 '만개의 레시피'에서 제공하는 추천 메뉴는 개인화 되어있지 않고 분류한 항목들을 보여주는 것에 그친다. 본 프로젝트에서는 요리에 관심이 있는 사람이 점점 많아지고 개인화 추천 서비스가 트랜드인 만큼 사용자가 선호하는 음식과 냉장고의 상태, 나이, 성별 등에 따라 군집화를 통해 사용자 맞춤형 요리 레시피 추천 서비스를 제공한다. 아직 대부분의 레시피 사이트는 재료의 종류나 조리법 등에 대해서만 구분이 되어있어서 사용자가 직접 선택 해야만 레시피를 찾을 수 있다. 하지만, 팀-6이 구축한 서비스를 사용하게 된다면 사용자는 어렵게 요리 레시피를 찾지 않아도 사용자가 좋아하고 현재 만들 수 있는 요리의 레시피를 추천 받을 수 있다. 현재 이러한 레시피 추천 서비스는 국내의 레시피 제공 사이트에서 찾아볼 수 있는 형태가 아니기 때문에, 기존의 레시피 추천 서비스와 사용자 만족도를 비교하기 보다는 팀-6이 제공하는 서비스의 만족도가 얼마나 높은지 조사하여 프로젝트를 평가한다. 서비스 만족도를 알기 위해서 사용자가 본 프로젝트의 서비스를 얼마나 많이 사용하고, 추천하는 레시피의 항목을 얼마나 많이 눌렀으며, 만족도 설문 조사를 통해 사용자가 만족할 만한 추천 레시피를 제공받았는지 조사한다. 위의 항목들을 종합하여 사용자의 만족도를 평가할 예정이다.

2.6. Project prototype

2.6.1. Are you building a web or mobile system?

팀 구성은 총 3명이다. 그 중 web project를 경험해본 1인과 mobile project를 경험해본 1인이 존재한다. 기존에는 크롬 확장 프로그램의 형태로 구현하여 사용자에게 제공하려고 했지만, 레시피를 고를 때는 컴퓨터를 켜기보다는 모바일 환경을 이용할 것이라는 생각이 떠올랐다. 또 사용성과 접근성을 고려하면 웹보다는 앱을 선택하는 것이 좋아 보였다. 모바일은 iOS를 개발할 수 있는 물리적인 환경이 조성되지 못하는 점에 따라 안드로이드만 지원할 계획이다. 개발 플랫폼은 Android Studio를 사용하고 언어는 Java를 사용한다. 서버는 Python Flask를 통해 구축한다

2.6.2. How much experience does your team have in building a such system?

본 프로젝트를 위해서 필요한 개발 단계는 다음과 같다.

- | | |
|---------------|-------------|
| 1. 웹 크롤링 | 5. 서버 구축 |
| 2. 데이터 전처리 | 6. 모바일 앱 제작 |
| 3. 모델 학습 | 7. 사용자 스터디 |
| 4. 추천 알고리즘 확립 | 8. 개발 내용 보완 |

데이터를 수집하기 위한 크롤링이나 데이터 전처리에 많은 경험이 있는 팀원은 없으며, 모델 학습 부분에서도 유능하게 할 수 있는 팀원은 없었다. 크롤링과 데이터 전처리, 모델 학습은 모두 공부를 하면서 진행해야 할 만큼 기초 지식이 부족했다.

서버 구축과 프론트엔드 페이지를 제작한 경험이 있는 팀원은 있으나 이 모두를 한명에게 맡기기에는 프로젝트 진행 기간과 완성도를 고려하였을 때 효율적이지 않기 때문에 서버 구축을 경험이 있는 팀원에게 맡겼고 앱 프론트엔드 구축은 잘 아는 팀원이 도와주고 같이 공부해 가면서 팀원들이 역할을 나누어 수행했다. 커뮤니티 구축은 이번 프로젝트 기간에 완성하기 어려울 것으로 예상되었지만 세 명이 서로 역할을 잘 분담하고 협력한 결과, 게시물 작성과 댓글, 사진 올리기까지 성공적으로 완료할 수 있었다.

또한, 도메인분석설계와 HCI 수업을 수강한 팀원들이 있어서 하나의 프로젝트를 완성해 본 경험이 있다. 따라서 개발 과정에서 발생할 수 있는 문제를 예상해서 미리 대비하고, 개발 단계마다 여러 가지 가능성 중 선택해야 할 때 최적의 선택을 할 수 있다. 필요하다면 모델링 언어를 사용해 다이어그램을 그려서, 우리가 정의해 둔 요구 사항을 더 구체적으로 다듬어 개발 과정에 도움을 줄 수 있다.

3. Affinity Diagram

Affinity diagram을 만드는 과정과 그에 따라 도출할 수 있는 기능과 서비스들에 대해 설명한다. 사람들이 이 시스템에 가질 것으로 예상되는 욕구들을 바탕으로 affinity notes를 작성했고, grouping과 group naming, adding experience patterns 과정을 통해 최종 affinity diagram을 완성하였다. 또 diagram에서 생각해 낼 수 있는 다섯 가지 기능과 서비스들을 설명하고, 그것들의 달성을 여부를 논한다.

3.1. Affinity Notes



[그림 2] Affinity notes

3.2. Grouping



[그림 3] Grouping

3.3. Group Naming



[그림 4] Group naming

3.4. Adding Experience Patterns



[그림 5] 최종 Affinity Diagram

최종 Affinity Diagram에서 정의한 기능과 서비스는 다음과 같다.

기능 및 서비스 1. 체계적인 레시피 분류 시스템

기능 및 서비스 2. 사용자 상황을 고려한 레시피 추천 시스템

기능 및 서비스 3. 다량의 레시피 정보 제공

기능 및 서비스 4. 자신의 레시피 공유 공간 제공

기능 및 서비스 5. 향상된 서비스 접근성 제공

본 프로젝트에서 우리는 위의 기능 및 서비스 중 1, 2번 시스템을 우선적으로 제공할 기능 및 서비스로 정의했다. 위의 두 가지가 완성된 후, 나머지 기능 및 서비스도 사용자 스터디를 통한

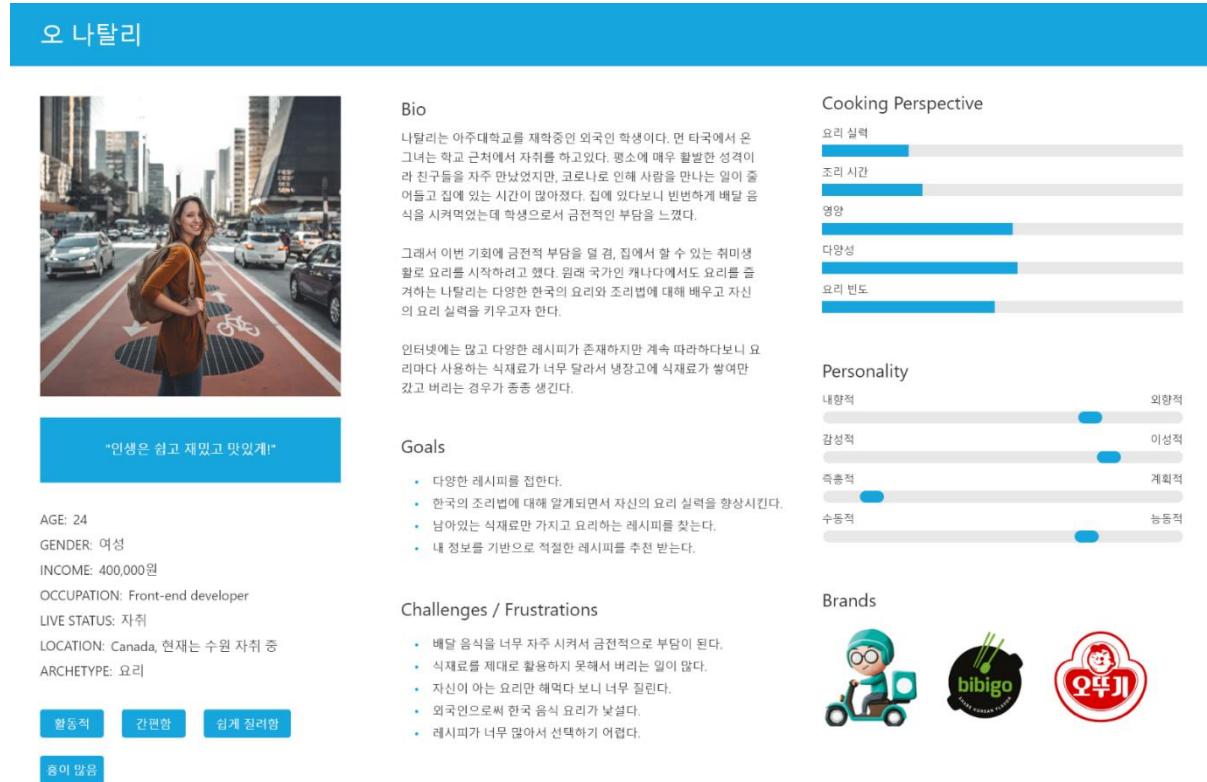
우선 순위를 정해 제공할 예정이었다.

최종 보고서를 쓰고 있는 현재 시점에서 돌이켜 보면, 다섯 가지 기능과 서비스 모두 어떤 형식으로든 어플리케이션에 잘 녹아들어 있다고 생각한다. 후술하겠지만 요리의 종류에 따라 레시피들을 분류했고, 추천 역시 머신러닝을 활용해서 개인별, 집단별 추천이 가능하고, 레시피 정보 역시 만개의 레시피의 레시피들 중 약 10만개 이상을 가지고 있다. 또 게시판 기능으로 자신의 레시피를 다른 사용자들과 공유할 수 있고, 계속해서 앱의 레이아웃을 고치고 에러 등을 잡아내면서 더 나은 사용성을 보인다.

4. Personas and Scenarios

설정한 target user들을 기반으로 시스템을 이용할 잠재적 사용자들에 해당하는 네 명의 페르소나들을 정의한다. 그에 따른 간단한 시나리오 역시 작성했다.

4.1. 요리가 하고 싶은 자취생 오 나탈리씨

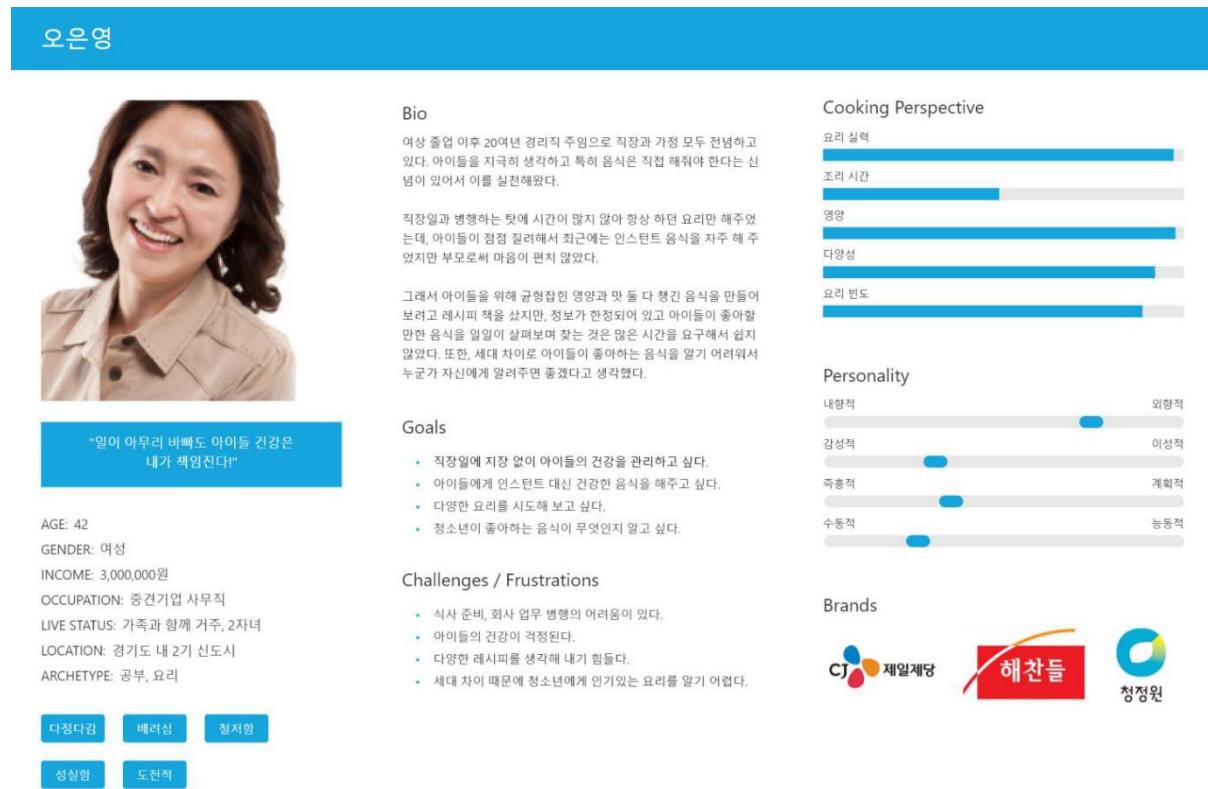


[그림 6] Persona – 오 나탈리

위 Persona는 1인 가구에서 다양한 음식들을 직접 요리해 먹지만 남는 식재료가 많아서 고민인 사람들을 대표한다.

- 외국인 학생으로 한국이 많이 낯설다.
- 한국의 배달음식이 편리하지만 가격이 너무 부담스럽다.
- 취미 생활로 요리에 대한 관심이 많다.
- 가리는 음식이 없기 때문에 다양한 음식을 요리해보고 싶다.
- 주변 마트에는 식재료를 대량으로만 판매하여 냉장고에 식재료가 쌓여가고 있다.
- 다양한 레시피를 얻을 수 있어 좋지만 이로 인해 선택의 어려움 있다.

4.2. 맛과 영양까지 고려해 요리를 하려는 주부 오은영씨

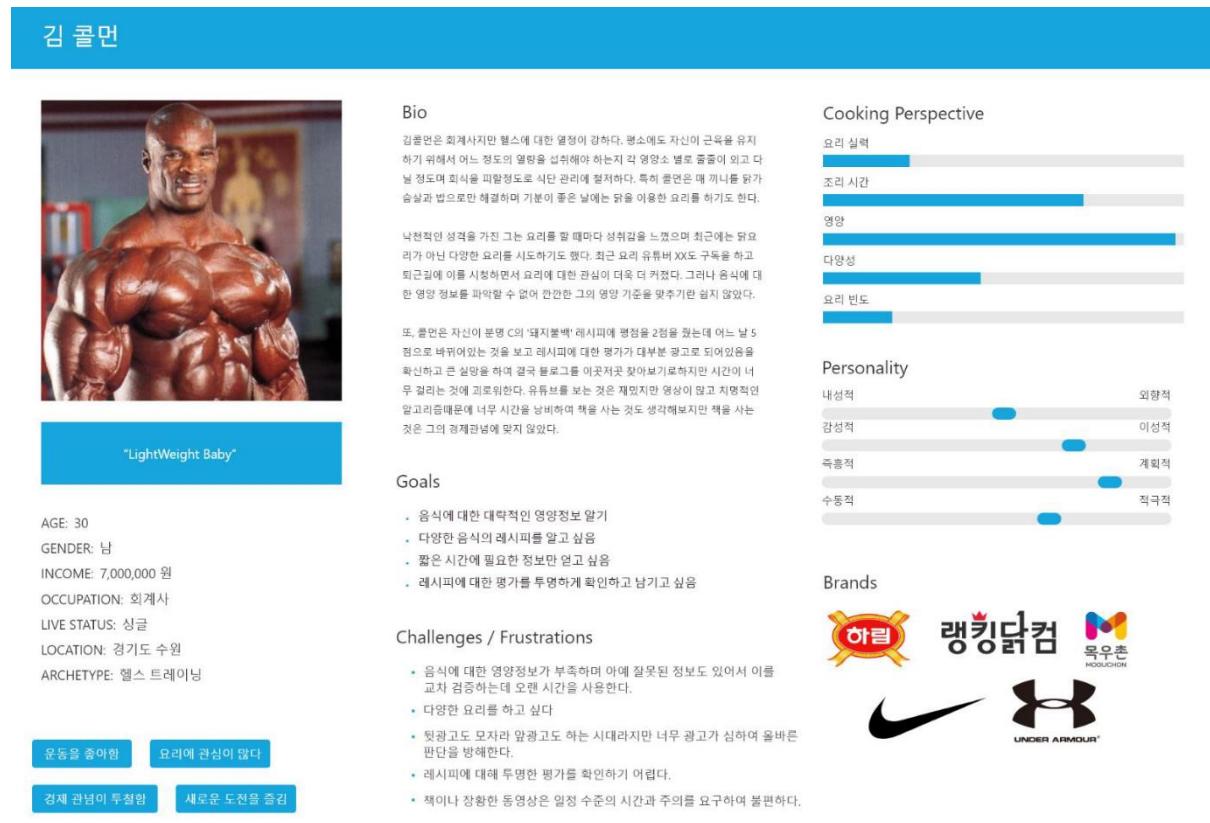


[그림 7] Persona – 오은영

위 Persona는 아이들에게 맛있고 다양한 음식을 직접 만들어 주고 싶지만 바빠서 그러지 못하는 직장인 겸 주부를 대표한다.

- 집안일과 직장생활을 병행하는 40대 주부이다.
- 자신의 일을 하면서 가족들의 끼니까지 챙겨 주는 것이 힘들다.
- 가족들이 제대로 챙겨 먹는지 걱정되고, 자신이 요리를 해 주지 못해 건강을 잃는 상황을 항상 걱정하고 있다.
- 아이들의 입맛과 영양을 고려한 레시피를 찾고 싶다.

4.3. 식단을 깐깐하게 지키는 직장인 김 콜먼씨



4.4. 대중들이 선호하는 레시피 정보를 얻고 싶은 요식업계 종사자 백종원씨

백종원



"요리 개발은 살면서 가장 즐거운 일"

AGE: 30
GENDER: 남
INCOME: 10,000,000 원
OCCUPATION: 요리 연구가
LIVE STATUS: 기혼
LOCATION: 서울 특별시
ARCHETYPE: 요리 연구

Bio

증원씨는 요리 연구가로 항상 새로운 요리를 어떻게 만들지 생각한다. 또한, 사람들에게 자신의 요리를 알려주기 좋아하며 얼마나 맛있는지 평가 받는 것을 즐긴다.

하지만, 새로운 요리를 혼자 개발하는 것은 한계가 있고 자신이 알고 있는 지식으로 만드는 요리는 점점 비슷해져서 고민에 빠져있었다. 더하여, 새로운 요리를 만들기 위해서는 요즘 사람들이 어떤 음식 트렌드를 따르고 있는지 알아야하는데 음식을 만들거나 먹을 때 무엇을 좋아할지 조사하기 어려워서 새로운 요리 개발이 더뎌지고 있다.

다양한 레시피가 있는 사이트에서도 사람들의 선호도나 트렌드를 파악하기는 어려웠기 때문에 사람들의 반응을 쉽게 얻을 수 있는 방법을 찾는 중이다.

Cooking Perspective

요리 실력	조리 시간	영양	다양성	요리 범도
90%	90%	85%	90%	90%

Goals

- 이전에 존재하지 않는 요리를 개발
- 나만의 레시피에 대한 평가를 받음
- 신메뉴를 개발할 때 사람들의 트렌드, 취향 파악

Personality

내성적	외향적
50%	50%

감성적	이성적
50%	50%

독창적	계획적
50%	50%

수동적	적극적
50%	50%

Challenges / Frustrations

- 음식 트렌드는 자주 변하며, 순쉽게 파악하기 어려움
- 창업이나 개발에 앞서 다양한 사람들에게 테스트하기 어려움
- 사람마다 가지고 있는 요리의 노하우를 공유하기 어려움
- 모든 음식의 기능성을 알고 있는 않기에 음식의 특성 파악

Brands



먹는걸 좋아함 새로운 요리법에 관심 많음

분석적
내침착

[그림 9] Persona – 백종원

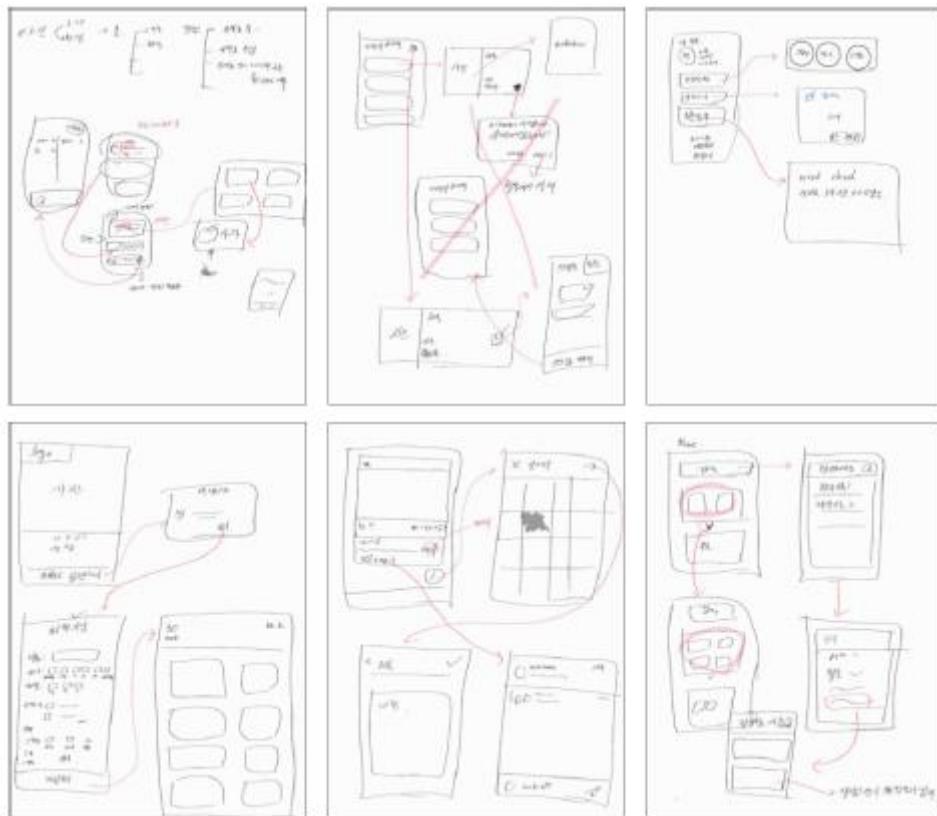
위 Persona는 다양한 레시피를 접해야 하고 대중들의 음식 트렌드를 파악하고 싶은 요식업계 종사를 대표한다.

- 혼자서 새로운 요리 개발하는 것이 한계에 달한 데 대한 고민이 있다.
- 요즘 사람들이 어떤 음식 트렌드를 따르는지 알고 싶어한다.
- 레시피 수가 많은 프로그램도 대중의 선호도 파악이 힘들었다.
- 사람들의 취향을 파악해서 좋은 평가를 얻기 위한 새로운 레시피 개발을 원한다.

5. Low-fi Prototype

이 장에서는 지금까지 페르소나와 시나리오를 작성하며 생각한 시스템의 기능들을 실제로 구현하기 위한 lowfi prototype을 작성한다. 우선 전반적인 밑그림, 즉 스케치를 한 뒤 전자에 그림을 그리고 접착시켰다.

5.1. Sketch

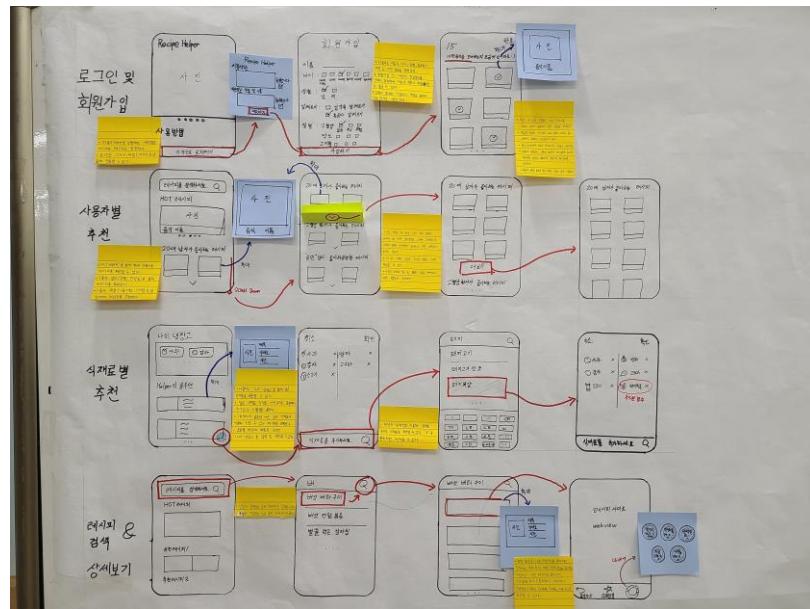


[그림 10] 1 차 Sketch

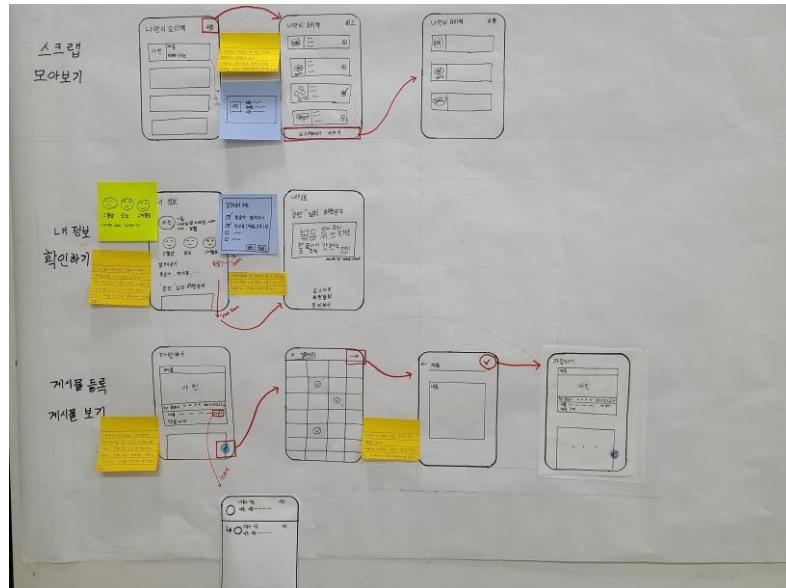
1차 계획, Sketch에서는 기능의 흐름이 어떻게 되는지 알아보고 버튼과 창의 대략적인 위치만 파악했다. 태블릿을 이용해 우리만 알아볼 수 있을 정도로 빠르게 그려낸 뒤, 스케치를 토대로 Low-fi prototyping을 진행하였다.

5.2. Low-fi Wireframe Prototype

앞서 그린 스케치에 기반해서 실제로 low fi prototyping을 진행했다. 우리는 기능별로 구간을 나누어서 누구나 한눈에 보고 이해할 수 있도록 쉽게 prototype을 만들었다.



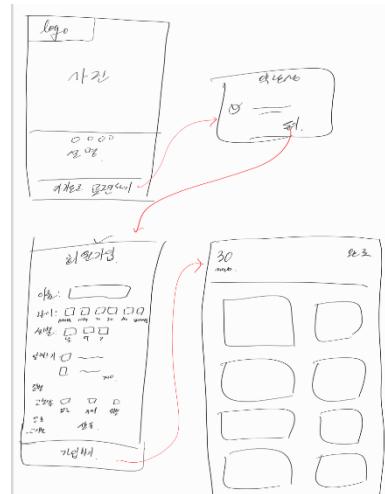
[그림 11] Recipe Helper low-fi prototype 1



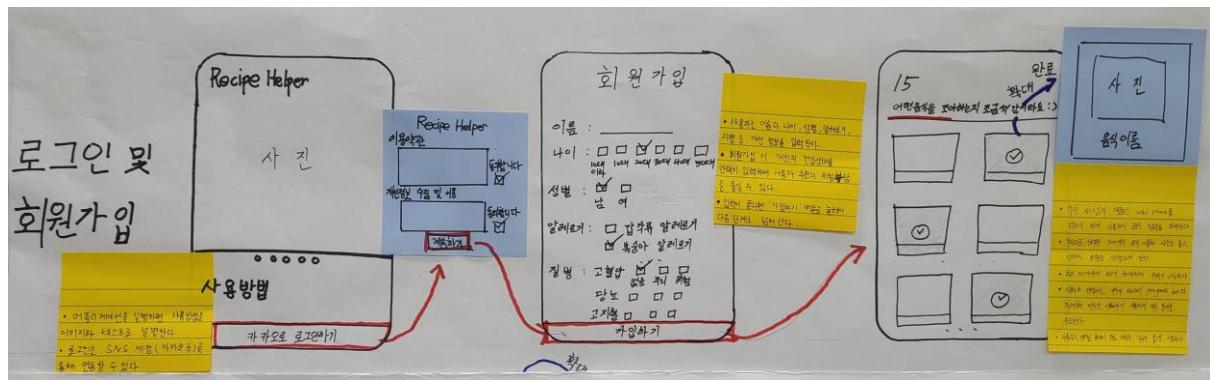
[그림 12] Recipe Helper low-fi prototype 2

Prototype의 전체적인 모습이다. 사진에서 확인할 수 있듯이 크게 7개의 기능으로 나누어진다. 차후의 개발은 모두 일곱 개의 기능들을 생각하면서 진행했다.

5.2.1. 로그인 및 회원가입



[그림 13] 로그인 및 회원가입 sketch

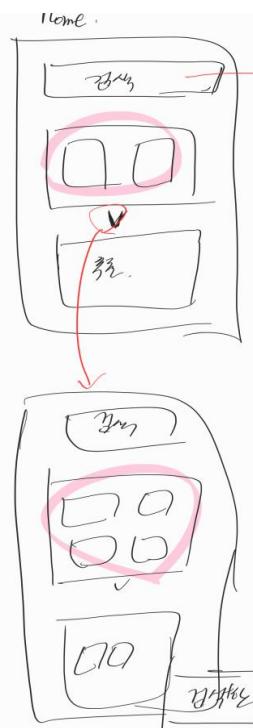


[그림 14] 로그인 및 회원가입 prototype

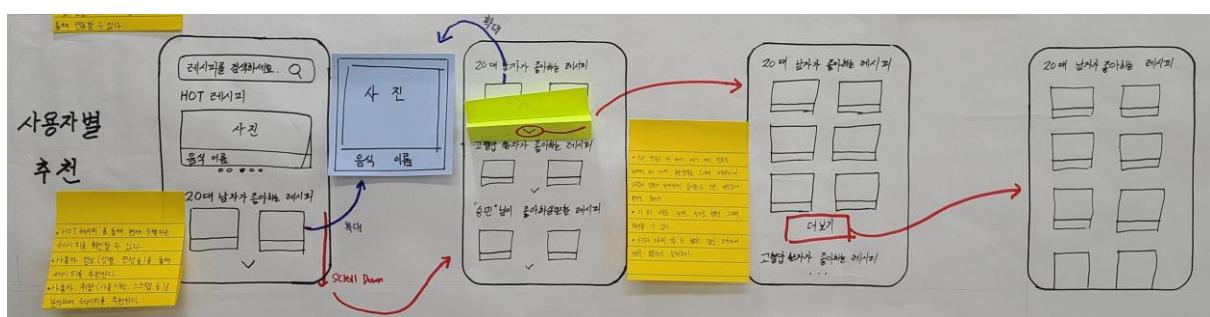
- Google계정 보다는 카카오 계정을 친숙하게 사용하는 사용자가 더 많을 것으로 판단하여 로그인은 카카오 계정을 이용하여 연동할 수 있도록 한다.
- 사용자는 회원가입시 카카오 계정 정보로 기본 제공되는 이름, 나이, 성별에 더해 알레르기, 질병 등 개인의 건강상태를 간략히 입력한다. 이러한 과정을 통해 추후에 제공할 추천 시스템에서 안전하게 사용자에게 레시피를 제공할 수 있다.
- 입력이 끝나면 '가입하기' 버튼을 눌러서 다음 단계로 넘어간다.

- 추천 시스템의 약점인 cold start를 피하기 위해 회원가입 시 사용자의 취향을 파악할 수 있는 단계를 진행한다.
- 사용자는 무작위로 나온 음식들 중 선호하는 음식을 선택한다. (최소 10개~최대 30개)
- 사용자가 선택하는 개수에 따라서 progress bar와 격려하는 멘트를 이용해서 사용자의 적극 참여를 유도한다.
- 사용자는 '완료'버튼을 누르면 선택 단계를 끝낼 수 있고 가입이 성공적으로 완료된다.

5.2.2. 사용자별 레시피 추천



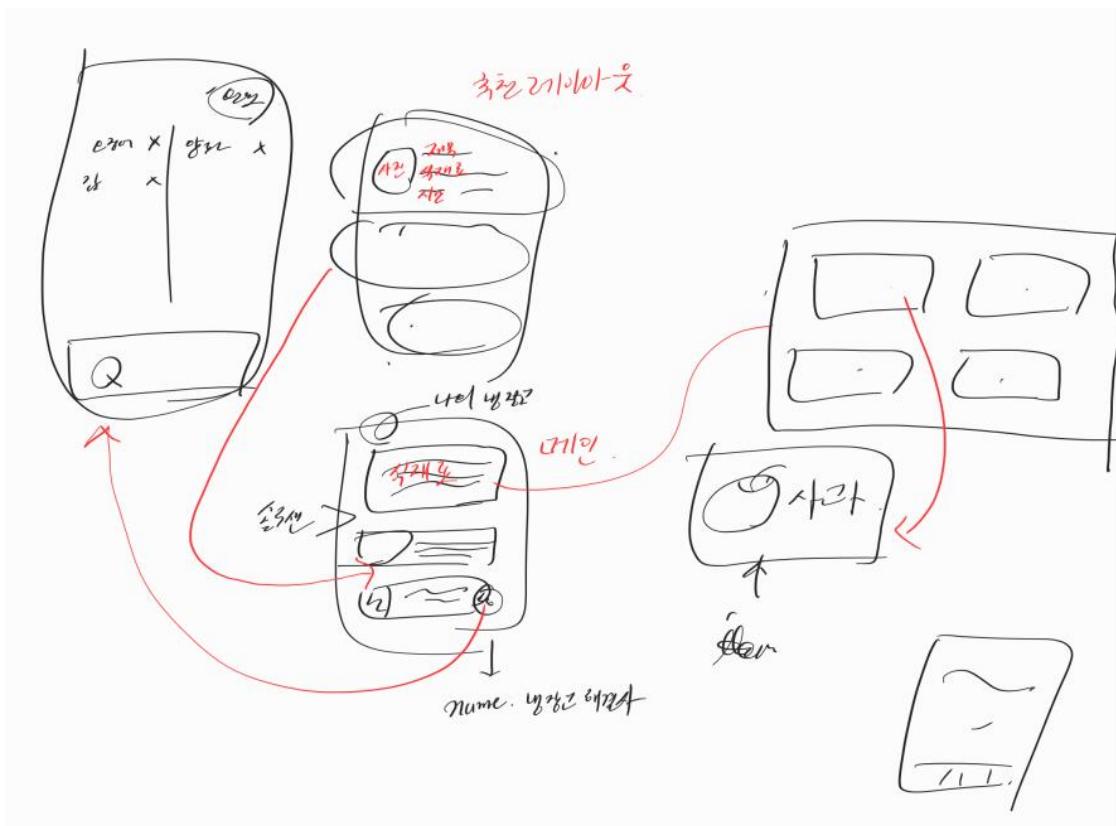
[그림 15] 사용자별 레시피 추천 sketch



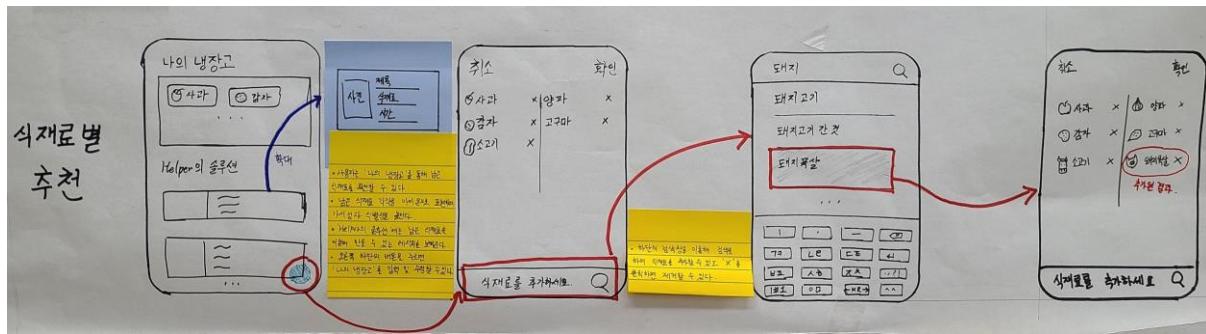
[그림 16] 사용자별 레시피 추천 prototype

- 'HOT 레시피'를 통해 현재 유행하는 레시피를 확인할 수 있다.
- 사용자 정보(성별, 연령 등)를 통해 레시피를 추천한다.
- 사용자 취향(사용 이력, 스크랩 등)을 분석하여 레시피를 추천한다.
- 추천 항목을 한 번에 여러 개 보여주는 것이 아니라, 추천 항목을 2개씩 배치하여 다양한 추천 항목을 볼 수 있고, 최소한의 정보로 많은 정보가 한꺼번에 올 때의 피로감과 혼란을 줄인다.
- '더 보기' 버튼을 누르면 추가로 항목이 2개씩 추가된다.
- 사진과 레시피 이름 등 필요한 정보를 요약하여 사용자에게 정보를 효율적으로 전달한다.

5.2.3. 식재료별 레시피 추천



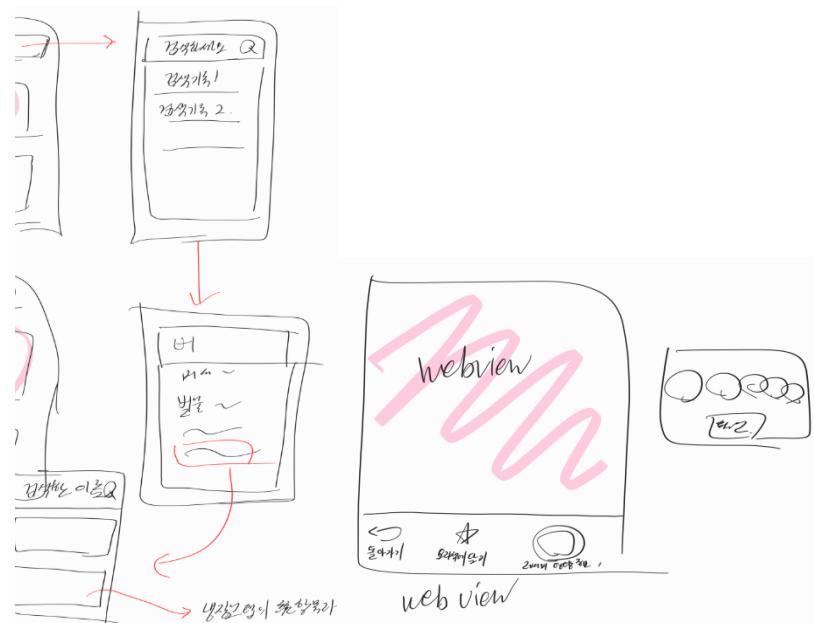
[그림 17] 식재료별 레시피 추천 sketch



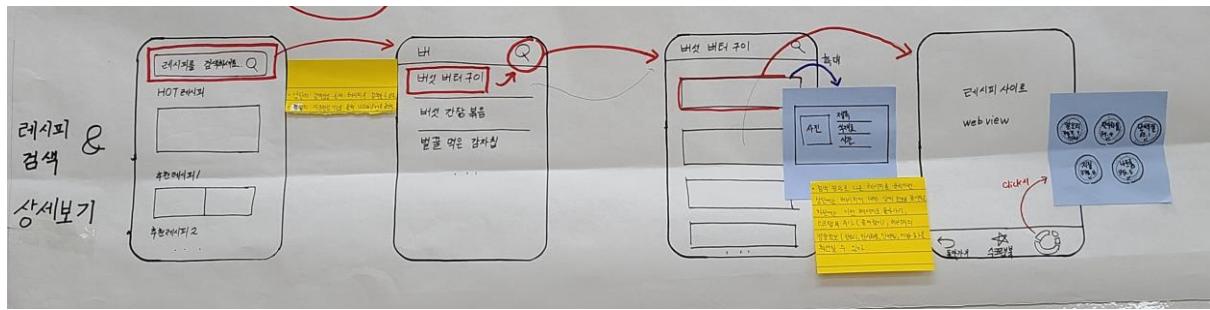
[그림 18] 식재료별 레시피 추천 prototype

- 사용자는 '나의 냉장고'를 통해 남은 식재료를 확인할 수 있다.
- 남은 식재료를 아이콘과 함께 표현하여 가시성과 식별성을 높인다.
- 'Helper의 솔루션'에서 버튼을 누르면 남은 식재료를 이용해 만들 수 있는 레시피를 보여준다.
- 오른쪽 하단의 버튼을 누르면 '나의 냉장고'를 입력 및 수정할 수 있다.
- 하단의 검색창을 이용해 검색하여 식재료를 추가할 수 있고, 식재료 기준 오른쪽에 위치한 'X'를 클릭하면 제거할 수 있다.

5.2.4. 레시피 검색 & 상세보기



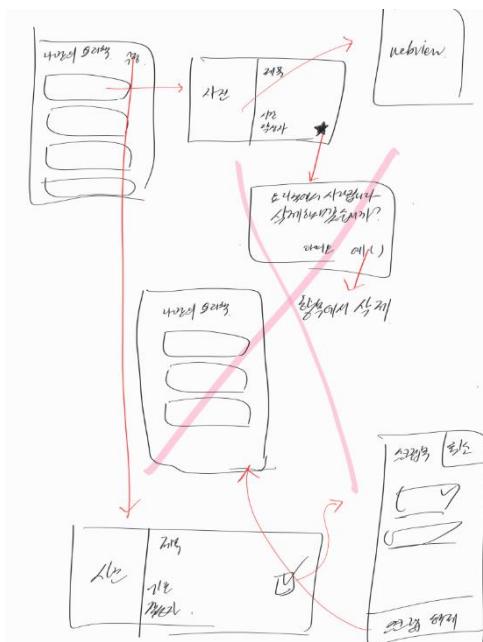
[그림 19] 레시피 검색 & 상세보기 sketch



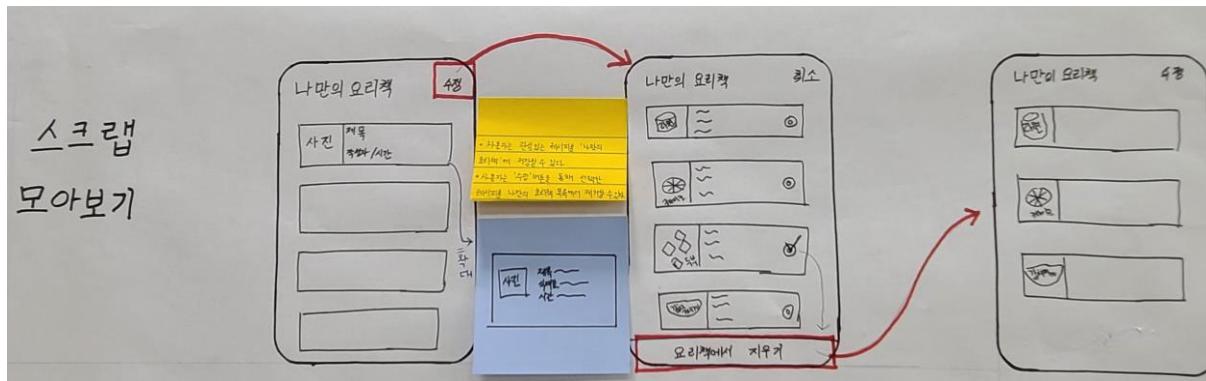
[그림 20] 레시피 검색 & 상세보기 prototype

- 상단의 검색창을 통해 레시피를 검색할 수 있다.
- 검색의 자동완성기능을 통해 사용자의 Usability를 높인다.
- 검색 결과로 나온 레시피를 클릭하면 상단에는 레시피에 대한 상세 정보를 표시하고 하단에는 Navigation bar를 이용해 이전 페이지로 돌아가기, 스크랩북 추가(즐겨찾기), 레시피의 영양정보(칼로리, 탄수화물, 단백질, 지방 등)을 확인할 수 있다. 실제 구현 시에는 원형 그래프 대신 간단히 숫자로만 표시했는데, 이는 원형 그래프를 도입하니 상대적으로 오히려 명시성이 떨어졌기 때문이다.

5.2.5. 스크랩 모아보기



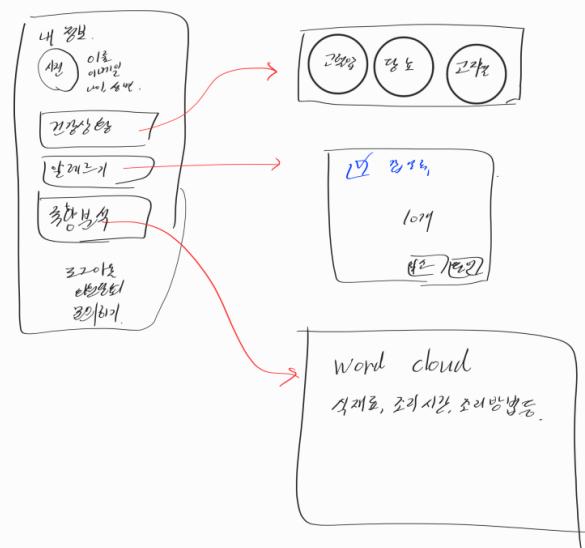
[그림 21] 스크랩 모아보기 sketch



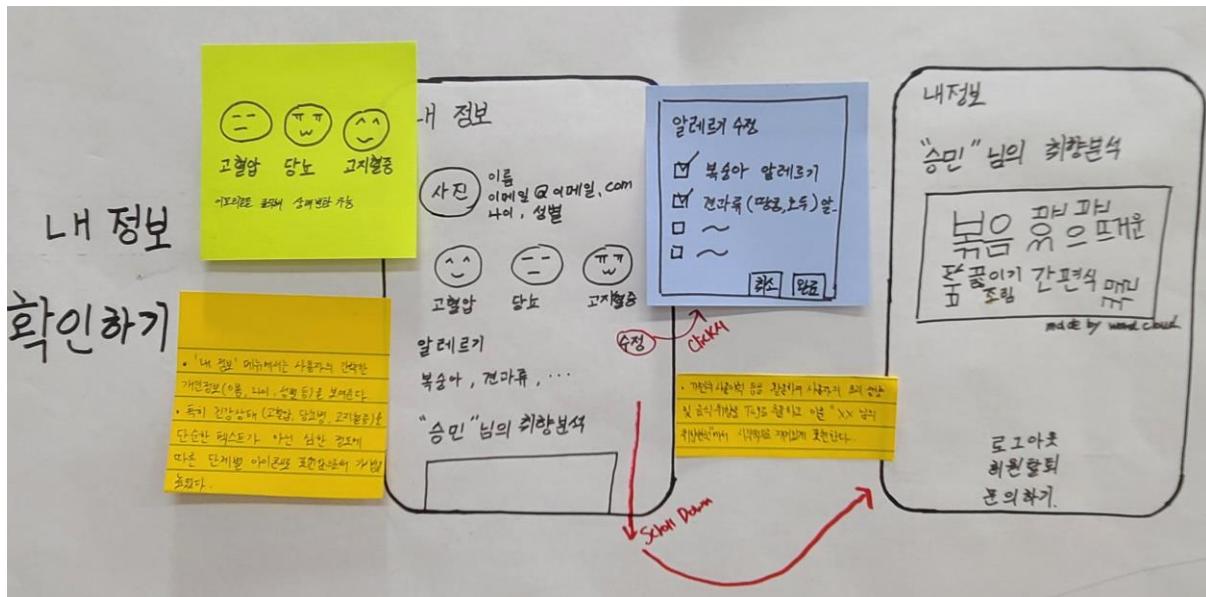
[그림 22] 스크랩 모아보기 prototype

- 사용자는 관심있는 레시피를 '나만의 요리책'에 저장할 수 있다.
- 사용자는 '수정'버튼을 통해 선택한 레시피를 나만의 요리책 목록에서 제거할 수 있다.

5.2.6. 내 정보 확인하기



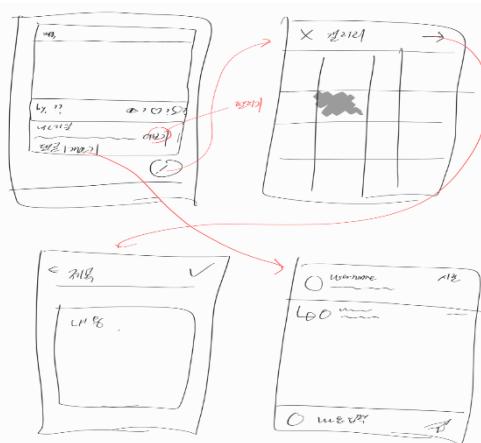
[그림 23] 내 정보 확인하기 sketch



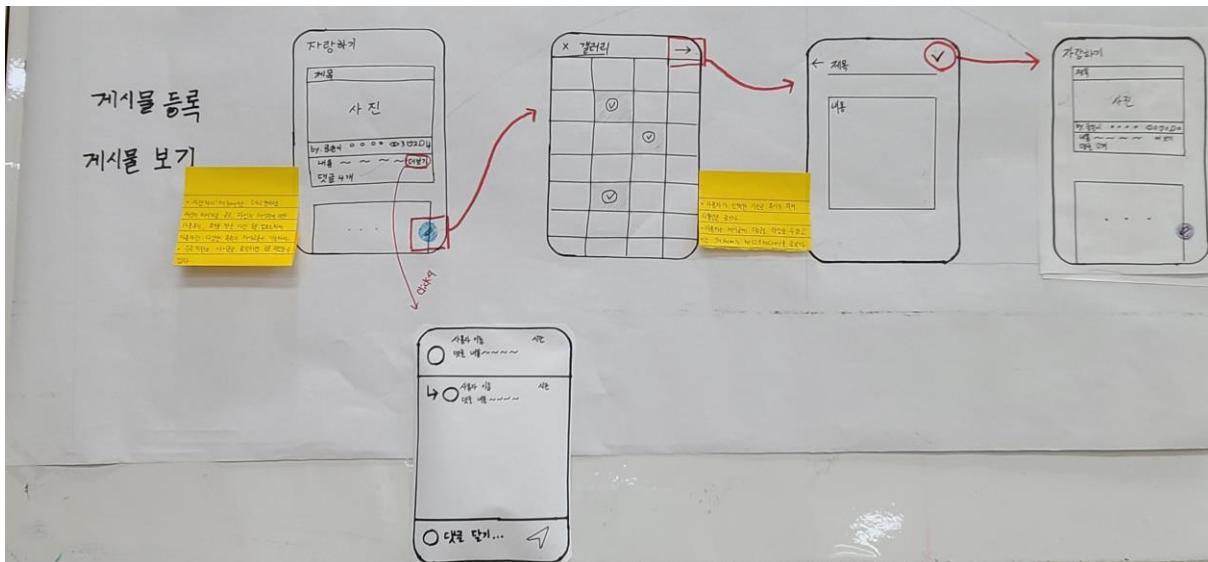
[그림 24] 내 정보 확인하기 prototype

- '내 정보' 메뉴에서는 사용자의 개인정보, 즉 이름과 나이, 성별 등을 보여준다.
- 건강상태는 세 가지 질병을 심각도에 따라 단계별로 아이콘을 표현해서 텍스트보다 가시성을 높일 수 있다.
- 사용자는 자신의 알레르기 상태를 수정할 수 있다.
- 개인의 사용이력 등을 활용하여 사용자의 요리 성향 및 음식 취향을 tag로 추출하고, 이를 시각적으로 재미있게 표현한다. 워드클라우드(wordcloud) 이미지를 통해서 자신도 몰랐던 자신의 미식 취향을 알 수 있을 것이다.

5.2.7. 게시물 등록 & 보기



[그림 25] 게시물 등록 & 보기 sketch



[그림 26] 게시물 등록 & 보기 prototype

- 간단한 게시판 기능을 '자랑하기'라는 탭으로 제공한다. 자신의 레시피를 다른 사람들에게 보여줄 수 있고, 타인의 레시피를 시도해 보고 사용후기를 올릴 수도 있다.
- 사진 업로드 시 선택된 사진들은 아이콘과 색깔들로 표시되어 자신이 선택한 사진의 식별성을 높일 수 있다.
- 작성이 끝났으면 오른쪽 상단의 'V' 모양 아이콘을 눌러서 작성을 완료한다. 그 후 자랑하기 메인 페이지로 돌아가서, 자신이 쓴 게시물이 업로드 되었음을 확인할 수 있다.
- 게시물에 쓰여진 댓글들을 확인하거나 자신이 댓글을 달 수 있다. 화면 하단에서 댓글을 쓰고 화살표 아이콘을 눌러서 업로드 할 수 있다.

6. Feedback from Group 1 and complementary measure

6.1. Feedback1

레시피 난이도 별로 사용자가 확인할 수 있으면 좋을 것 같다.

만개의 레시피 크롤링 데이터에 적혀 있는 난이도는 레시피를 업로드한 유저의 자의적 기준이라 신빙성이 떨어진다. 따라서 검증되지 않은 난이도보다는 레시피를 보고 사용자가 직접 판단해, 재료를 더하거나 빼는 식으로 난이도를 조절하거나 다른 레시피를 선택하는 것이 더 나을 것이다. 또한, 90% 이상이 '보통' 정도의 난이도를 가지고 있기 때문에 난이도 별로 보여주는 것은 큰 의미가 없다고 판단하였다.

6.2. Feedback2

레시피를 공유할 때 포인트를 주고, 포인트를 쓸 수 있게 만들어 사용자가 사용하고 싶게 만들면 좋을 것 같다.

우리는 레시피를 공유하는 행위가 '위키피디아'처럼 누구나 쉽게 작성하고 의견을 제시하여 백과사전을 만드는 것이지 수익성을 위해서라고 생각하지 않았다. 때문에 따로 리워드 시스템은 만들지 않았다.

6.3. Feedback3

머신 러닝 기법에 대해 구체화할 필요가 있다.

제한된 발표시간 내에 필요한 내용을 담다 보니 피드백 받을 저 당시에는 미처 설명을 못했다. 사용자의 취향을 분석하여 콘텐츠 기반의 추천시스템과 연령, 성별 등 개인 정보에 기반하여 협업 필터링의 사용자 기반 추천 시스템을 융합하여 활용할 예정이었다.

자세한 머신 러닝 기법은 다음 7장에서 설명한다.

6.4. Feedback4

영양 정보가 없는 레시피에 대해 다른 방법론을 적용하여 영양 정보를 구할 수 있으면 좋겠다.

역시 저 당시에는 발표 시간내에 구체적으로 설명을 못 했는데, 식약처 DB에 있는 레시피를 구성하는 각 재료의 영양성분을 합하여 레시피의 영양성분을 제공하려고 한다.

6.5. Feedback5

남은 식재료로 만들 수 있는 레시피 솔루션을 보여줄 때 추천 레시피에 없는 재료가 있다면 없는 것이 무엇인지 보여주면 좋을 것 같다.

저 당시에는 구현하지 않아 보여줄 수 없었지만, 최종 앱 개발에 반영되었다.

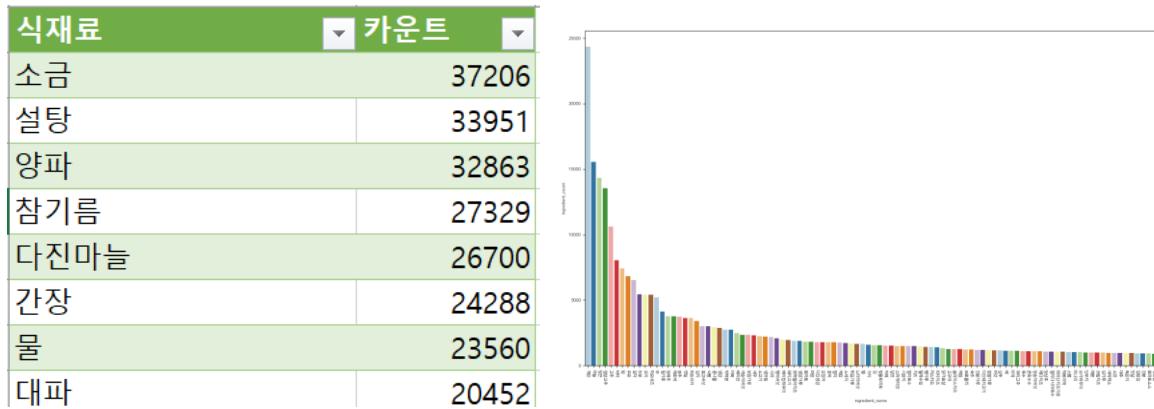
7. Machine Learning Model Description

데이터의 수집 과정, 데이터 전처리, Project의 필수 사항인 ML Model이 시스템에서 어떤 기능으로 사용되는지, 사용된 모델의 원리와 Recipe Helper가 이를 어떤 식으로 구현했는지 전체 Model을 설명하는 그림과 함께 Model의 개발 과정을 단계별로 설명하며 그에 대한 구체적인 이유로 설명한다. 또한 실제로 구현된 모델의 성능과 한계점, 이를 발전시킬 수 있는 방안에 대해서 서술한다.

7.1. Data Description (Sample, Feature)

초기 Sample Size: 약 140,000개

국내에서 레시피에 대한 정보를 가장 많이 가지고 있는 '만개의 레시피' 사이트에서 크롤링한 결과로 총 14만 개의 데이터를 확보했다.



[그림 27] (좌) Feature 의 사용량 순으로 정리한 모습, (우) Feature 의 사용량 그래프

초기 Feature의 전체 개수: 72,000개

레시피라는 각 데이터를 설명할 수 있는 것은 식재료이기 때문에 레시피를 구성하는 식재료를 데이터를 수집한 시점에서 Feature로 정했다. 추후 전처리 작업을 위해 [그림 27]과 같이 레시피에서 사용된 재료의 이름과 레시피에 사용된 횟수를 같이 구성했다.

7.2. 예측하고자 하는 y 값 설명

User_ID \ Recipe_ID	12345	12346	12347	12348
User_ID				
1	1	?	1	?
2	1			1
3		1		1
4			1	
...				

[표 2] Recipe_ID – User_ID click 표

Recipe_id	Predict_Value
12345	0.96
12346	0.88
12347	0.92
12348	0.77
...	...

[표 3] Recipe ID 별 유사도

최종적으로 예측하고자 하는 y값은 사용자가 어떤 레시피를 선호할 만한 레시피이다. [표 2]는 사용자가 해당 레시피를 클릭한 경우 1을 나타내고 그렇지 않은 경우 0을 나타낸다. 이는 사용자의 개인정보와 사용이력 등을 활용하여 사용자간 유사도를 측정한 후 유사한 성향의 사용자를 찾아 추천하는 사용자 기반 협업 필터링 방식(User based Collaborative Filtering)과 레시피간 유사도를 측정하여 사용자가 경험한 레시피와 비슷한 것을 찾아주는 콘텐츠 기반 필터링(Content Based Filtering)방식을 결합하여 사용하는 하이브리드 협업 필터링 방식을 통해 각 레시피마다 예측 값(Predict_Value)이 산출된다. 이 예측 값이 높은 순으로 사용자가 선호할 만한 레시피(y = 레시피의 id)로 추천된다.

7.3. Data Preprocessing

Sample

- Null이나 중복되는 값 등과 같이 사용할 수 없다고 판단되는 레시피를 모두 제거.
- 레시피의 정보가 부정확하다고 판단되는 2개 이하의 식재료를 가진 레시피를 모두 제거.
- 레시피의 정보라기 보다는, '감자 삶는 법'과 같이 식재료를 손질하는 방법 모두 제거.
- Feature로 사용될 식재료를 Preprocessing하면서 통폐합된 식재료의 이름을 레시피 단위로 접근하여 update. ex) 흥고추 → 고추

- 결과적으로 약 14만 개에서 10만 개로 sample 축소.

Feature



[그림 28] Data Preprocessing 예시

- 식재료 중 총 100회 미만으로 사용된 식재료 제거.
- 식재료 중 비 문법적인 단어(띄어쓰기, 비표준어 사용 등)를 표준어에 맞춰 통합.
- 식재료 중 비슷한 역할을 하거나 여러 명칭으로 불리는 단어를 하나로 통합.
- 결과적으로 약 72,000개의 Feature를 총 350개로 축소.

<전처리 결과>

	id	class	마늘	소금	간장	양파	설탕	대파	물	후추	...
19558	6908484	27	1	0	0	0	0	0	5	0	...
21279	6853428	27	0	0	0	0	0	0	0	0	...
29705	6923213	27	1	1	1	0	0	0	5	0	...

[그림 29] 레시피의 DataFrame 전처리 결과


```

{'굽기_빵': 0,
'굽기_구이': 1,
'굽기_떡갈비': 2,
'굽기_떡꼬치': 3,
'굽기_디저트': 4,
'굽기_마카롱': 5,
'굽기_베이컨말이': 6,
'굽기_샌드위치': 7,
'굽기_스테이크': 8,
...

```

[그림 32] 최종 클래스 레이블링 결과

두 단계에 걸친 클래스 레이블링의 결과 총 413개에서 100개로 축소할 수 있었다.

7.5. 사용한 모델 알고리즘: 하이브리드 추천 시스템

레시피 유사도 측정 알고리즘

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

[그림 33] Euclidean Distance

레시피의 유사도를 측정하기 위해서는 KNN 알고리즘과 같이 항목간 유사도를 구할 수 있는 알고리즘을 사용한다. 각 레시피간 유사도를 Euclidean Distance 방식을 통해 거리를 구하는데, 거리가 짧을수록 두 레시피는 사용되는 식재료가 비슷하기 때문에 두 레시피는 유사하다고 할 수 있다.

SVD 행렬 분해 기법

$$\begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & \\ \vdots & \vdots & \ddots & \\ x_{m1} & & & x_{mn} \end{pmatrix}_{m \times n} \approx \begin{pmatrix} u_{11} & U & \dots & u_{1r} \\ \vdots & \ddots & & \\ u_{m1} & & & u_{mr} \end{pmatrix}_{m \times r} \begin{pmatrix} s_{11} & 0 & \dots \\ 0 & \ddots & \\ \vdots & & s_{rr} \end{pmatrix}_{r \times r} \begin{pmatrix} v_{11} & V^T & \dots & v_{1n} \\ \vdots & \ddots & & \\ v_{r1} & & & v_{rn} \end{pmatrix}_{r \times n}$$

[그림 34] SVD

SVD(SVD: Singular Value Decomposition)는 차원 축소 기법을 사용하여 사용자의 특징과 아이템의 특징, 그리고 이 두 가지를 대표하는 대각행렬을 추출한다. 대각 행렬의 특이값으로 차원을 축소(데이터의 양을 줄임)할 수 있다는 장점을 가지고 있다. 행렬 분해 기법을 통해 아직 평가하지 않은 데이터에 대해서 평균값을 이용해 결측치를 채운 뒤 SVD를 이용해 점수를 예측한다. 요컨대, 사용자 x 레시피 DataFrame이 주어지면 특정 사용자가 아직 평가하지 않은 레시피들에 대한 예측값(클릭할 것인가?)을 계산하고 이 중 점수가 높은 순서대로 레시피를 추천한다는 것이다.

(개인)사용자 맞춤형 추천 알고리즘

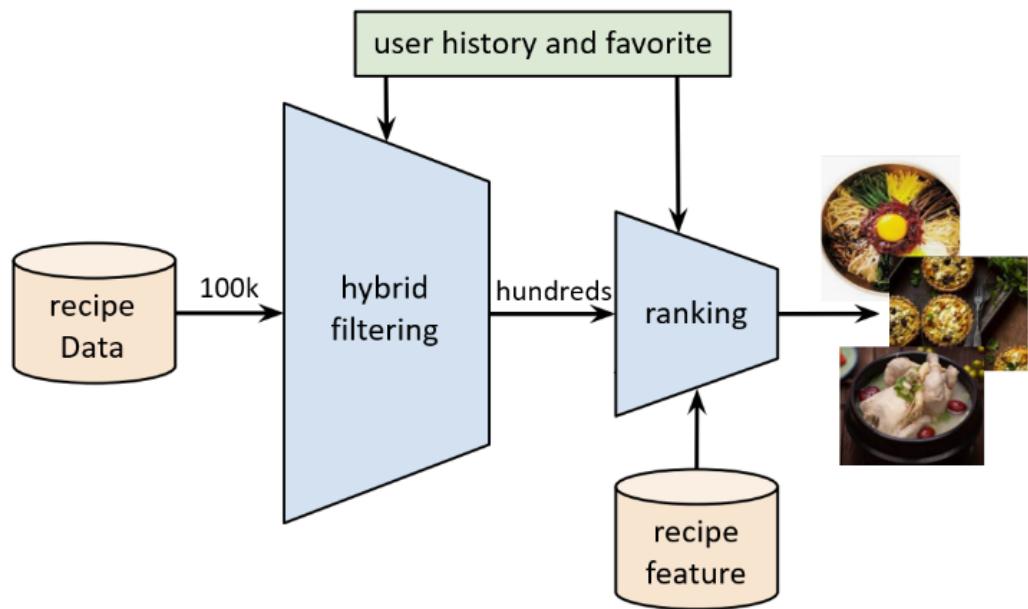
- 1 단계: 콘텐츠 기반 협업 필터링 기법을 사용한다. 회원 가입시 사용자에게 최소 10 가지의 음식을 선택하도록 하여 미리 사용자 x 레시피 DataFrame 값을 채웠기 때문에 cold start 의 문제는 없다. 이 데이터를 기준으로 미리 labeling 한 그룹(food_group) 중 가장 많은 수를 차지하고 있는 그룹을 최대 3 개 선택하되, 가장 최근에 선택한 레시피를 기준으로 그 그룹에서 임의로 100 개씩 선택한 후, 레시피간 유사도를 구하여 그중 상위 30 개씩을 추천 리스트로 만든다. 이를 요약하면 다음과 같다.
 1. 사용자가 가장 많이 클릭한 food_group 중 상위 3 개 선택
 2. 해당 food_group 내에서 임의의 레시피 100 개씩 선택
 3. 선택된 food_group 중 가장 최근에 선택한 레시피를 기준으로 추출한 100 개와의 유사도를 Euclidean Distance 를 통해 구함.
 4. 가장 가까운 레시피를 30 개씩 골라서 추천 리스트를 만든다.
- 2 단계: SVD 를 활용한 잠재 요소 기반 협업 필터링 기법을 통해 사용자가 아직 평가하지 않은 레시피에 대한 예측값을 출력하면 이를 기준으로 사용자가 적기 때문에 우선 상위 10 개만을 선택하여 추천 리스트를 만든다.
- 3 단계: 1 단계와 2 단계에서 생성한 추천 리스트를 취합하여 최종적으로 사용자에게 추천한다. 이때 초기 사용자의 수가 적기 때문에 콘텐츠 기반 협업 필터링의 결과의 우선순위가 더 높게 한다.

(연령/성별) 맞춤형 추천 알고리즘

사용자의 정보를 기준으로 사용자와 같은 연령/성별인 사용자들의 정보를 모아서 입력값으로 SVD 협업 필터링을 통해 사용자가 아직 평가하지 않은 레시피에 대한 예측값을 출력하고, 이를 기준으로 상위 10개만을 선택하여 추천 리스트를 생성하여 사용자에게 추천한다.

위와 같이 사용자의 사용이력, 개인정보 등을 토대로 다양한 추천폭을 가지는 요리 레시피를 추천하는 컨텐츠 필터링과 협업적 필터링의 장점을 포괄하는 하이브리드 필터링(Hybrid Filtering)을 적용한 추천 시스템을 사용한다. 이러한 하이브리드 기법을 통해 Collaborative Filtering과 Content Based Filtering의 장단점을 서로 보완하여 사용자에게 보다 더 다양하고 풍부한 레시피를 추천해줄 수 있다.

7.6. 전체 모델 구조를 설명하는 그림 작성



[그림 35] Recipe-Helper Hybrid Collaborative Filtering

7.7. 표에 관한 설명

	Ingred_1	Ingred_2	Ingred_3	Ingred_4	...
Recipe_1	1	0	0	1	
Recipe_2	0	0	0	1	
Recipe_3	0	1	0	1	
...					

[그림 36] 레시피 – 재료 DataFrame

	recipe_1	recipe_2	recipe_3	recipe_4	...
user_1	1	0	1	0	
user_2	0	0	1	1	
user_3	1	1	0	0	
...					

[그림 37] 사용자 – 레시피 DataFrame

	class_1	class_2	class_3	class_4	...
user_1	10	0	1	4	
user_2	6	5	3	7	
user_3	1	1	8	0	
...					

[그림 38] 사용자 – Food_group DataFrame

	class_1	class_2	class_3	class_4	...
user_1	43131		39381	12345	
user_2		49928	48288	23201	
user_3	38239	83912	23012		
...					

[그림 39] 사용자 – Latest_Clicked_Recipe_by_class

- 레시피 - 재료 행렬: 콘텐츠 기반 추천 알고리즘(Content Based Filtering)을 위해 사용한다. 레시피의 구성 재료를 Feature로 사용하여 이와 유사한 레시피를 찾기 위해 KNN 알고리즘을 사용하여 레시피간 distance를 측정한다. 이때 distance가 짧을수록 두 레시피가 유사함을 의미한다.
- 사용자 - 레시피 행렬: 사용자의 레시피에 대한 관심도(즐겨찾기 여부, 클릭 수, 평점 등을 종합하여 계산한 것)를 통해 유사한 사용자들을 찾아내 추천할 레시피를 찾기 위해 사용자 기반 협업 필터링 알고리즘(User-Based Collaborative Filtering)을 사용한다.
- 사용자 – food_group 행렬: 전처리 단계에서 미리 정해 놓은 100가지의 food group 중 어떤 food group을 사용자가 선호하는지 파악하기 위해서 사용자가 food_group 별로의 관심도를 나타낸다. 이를 통해 추천시 위 행렬에서 가장 큰 값을 가진 3개의 group을 선택해서 추천 리스트를 만들 수 있게된다.
- 사용자 – Latest_Clicked_Recipe_by_class 행렬: 이 행렬에는 각 food_group마다 가장 최근에 클릭한 레시피의 id를 저장한다. 이를 통해 추천리스트를 만들 때 가장 높은 관심도를 가진 상위 3개의 food_group의 가장 최근에 클릭한 레시피를 기준으로 유사도를 계산할 수 있다.

7.8. 성능결과

```
0      0.000000
16     2.449490
5      2.828427
37     3.162278
10     3.316625
36     5.477226
46     5.477226
30     5.656854
41     5.656854
38     5.744563
23     5.830952
32     7.211103
1      7.348469
21     7.416198
42     7.416198
44     7.416198
15     7.483315
9      7.549834
4      7.615773
24     7.681146
Name: 0, dtype: float64
```

[그림 40] Content Based Filtering Result

[그림 40]은 사용자가 가장 많은 관심을 보이는 상위 3개의 food_group 중 하나의 food_group에 대한 결과를 예시로 보여준다. 가장 최근에 클릭한 레시피를 기준으로 Euclidean Distance 방식으로 유사도를 계산하였고, 상위 20개만을 출력했다.

	rating										
recipe_id	1566393	5478393	6840149	6846025	6847276	6854399	6854799	6863696	6875490	6882477	...
user_id	1	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
8	0.0	2.0	0.0	2.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	1.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
13	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
15	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0

[그림 41] 실제 사용자 – 레시피 DataFrame

[그림 41]은 사용자의 각 레시피에 대한 관심도를 나타낸 DataFrame이다. 이를 SVD를 활용한 Latent Factor Collaborative Filtering Model의 input으로 사용하며, 위 결과로 특정 사용자에 대한 각 레시피에 대한 관심도에 대한 예측값이 계산되며 그 결과는 아래와 같다.

```

user: 9      item: 1566393    r_ui = 0.00    est = 1.01    {'was_impossible': False}
user: 9      item: 5478393    r_ui = 0.00    est = 1.26    {'was_impossible': False}
user: 9      item: 6840149    r_ui = 0.00    est = 1.34    {'was_impossible': False}
user: 9      item: 6846025    r_ui = 0.00    est = 1.62    {'was_impossible': False}
user: 9      item: 6847276    r_ui = 0.00    est = 1.05    {'was_impossible': False}
user: 9      item: 6854799    r_ui = 0.00    est = 1.48    {'was_impossible': False}
user: 9      item: 6882477    r_ui = 0.00    est = 1.22    {'was_impossible': False}
user: 9      item: 6884795    r_ui = 0.00    est = 0.92    {'was_impossible': False}
user: 9      item: 6885470    r_ui = 0.00    est = 1.05    {'was_impossible': False}
user: 9      item: 6889813    r_ui = 0.00    est = 1.43    {'was_impossible': False}
user: 9      item: 6907924    r_ui = 0.00    est = 1.10    {'was_impossible': False}
user: 9      item: 6908683    r_ui = 0.00    est = 1.12    {'was_impossible': False}
user: 9      item: 6913076    r_ui = 0.00    est = 1.17    {'was_impossible': False}
user: 9      item: 6917899    r_ui = 0.00    est = 1.01    {'was_impossible': False}
user: 9      item: 6920001    r_ui = 0.00    est = 1.49    {'was_impossible': False}
user: 9      item: 6921349    r_ui = 0.00    est = 1.54    {'was_impossible': False}
user: 9      item: 6921429    r_ui = 0.00    est = 1.32    {'was_impossible': False}
user: 9      item: 6925795    r_ui = 0.00    est = 1.60    {'was_impossible': False}

```

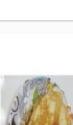
[그림 42] Latent Factor Collaborative Filtering

[그림 42] Latent Factor Collaborative Filtering의 결과이다. Item은 레시피의 id, EST(Estimate)는 특정 사용자(user_9)가 경험하지 않은 레시피들에 대해서 다른 사용자들의 데이터를 사용하여 계산한 예측값이다.

```
[ [ '6846025', 1.6151829101426958],
  [ '6925795', 1.600636361408337],
  [ '6921349', 1.5437845976751237],
  [ '6920001', 1.4917635106696634],
  [ '6854799', 1.4786937438539836],
  [ '6889813', 1.4253614846770788],
  [ '6840149', 1.3393570421995389],
  [ '6921429', 1.3206036353812824],
  [ '5478393', 1.2554145088721635],
  [ '6882477', 1.2197885267029773] ]
```

[그림 43] [그림 42]의 결과를 내림차순으로 정렬한 후 상위 10개를 추출한 결과

[그림 42]의 결과 중 예측값이 가장 높은 상위 10개의 레시피의 id를 추출한다. 그 결과는 [그림 43]과 같다. 이 결과는 현재 사용자들의 특성에 의해 결정되기 때문에 잠재 요인 협업 필터링의 특성상 결과에 대한 이유를 분석할 수 없다.

'허범수'님이 좋아하실 만한 레시피	'허범수'님이 좋아하실 만한 레시피	'허범수'님이 좋아하실 만한 레시피	'허범수'님이 좋아하실 만한 레시피
 <h3>간단한데 월리티 높은 생 초코 타르트</h3> <p>초콜릿 생크림을 넣탕 코카아루 by C Channel 디저트 30분 이내 초급</p>	 <h3>링파우더 설탕</h3> <p>by 디자인한스푼 디저트 120분 이내 초급</p>	 <h3>시나몬 크럼블 케이크</h3> <p>계란 설탕 박력분 소금 버터 계피가루 베이킹파우더 꿀 by 분홍들고래 디저트 60분 이내 고급</p>	 <h3>시원한 체리무스케이크</h3> <p>플레인요거트 버터 크림치즈 바닐라 익스트랙 레몬즙 생크림 설탕 by 카푸치노 디저트 2시간 이상 중급</p>
 <h3>치즈와 불닭의 만남 [불닭피자]</h3> <p>토마토 스파게티 칠리소스 파인애플 체다치즈 양파 파프리카 단호박 달... by 달집아들 디저트 30분 이내 아무나</p>	 <h3>와플버거[리큅 와플메이커 레시피]</h3> <p>베이컨 올리브 마요네즈 스테이크소스 피클 물 버터 토마토... by 리큅 디저트 30분 이내 초급</p>	 <h3>커피향이 솔솔 모카 쉬폰케이크(모카쉬폰케이크) 만들기</h3> <p>우유 식용유 계란 시럽 박력분 설탕 소금 물 생크림 설탕 베이킹파우더... by 열기왕고chiiiii 디저트 90분 이내 중급</p>	 <h3>또띠아피자 햄야채가 들어있는 아이들의 영양간식 한끼...</h3> <p>햄 양파 또띠아 양송이버섯 퍼프리카 토마토소스 단호박 by 등우등학맘 디저트 30분 이내 초급</p>
 <h3>카레피자</h3> <p>모짜렐리치즈 물 소금 드라이이스트 카레 올리브유 설탕 by 예예 디저트 90분 이내 아무나</p>	 <h3>함박스테이크♡</h3> <p>계란 소고기 양파 소금 후추 물 병어 류 식용유 되지고기 케밥 by 일년내내파티 스테이크 60분 이내 초급</p>	 <h3>퐁신풍신한 제누워즈 만들기 : 케이크시트 실패없이 만드는...</h3> <p>계란 박력분 버터 설탕 by 만개의레시피 디저트 90분 이내 초급</p>	 <h3>코코아 아몬드 쿠키</h3> <p>계란 박력분 아몬드 버터 초코칩 설탕 코코아가루 by Whalemilk 디저트 60분 이내 초급</p>
 <h3>야식으로 초간단 또띠아피자, 콘치즈옥수수피자</h3> <p>모짜렐라치즈 옥수수 크림치즈 또띠야 디저트 15분 이내 초급</p>	 <h3>치킨스테이크(집에서도 맛있게 되는~)</h3> <p>달고기 전분 양파 스테이크소스 소금 후추 율리고당 파프리카 by 비巴斯밥 스테이크 15분 이내 초급</p>	 <h3>수풀레펜케이크, 펜케이크, 핫케이크.</h3> <p>우유 버터 계란 박력분 바닐라아이스크림 식용유 베이킹파우더 설탕 디저트 15분 이내 초급</p>	 <h3>녹차 볼 쿠키</h3> <p>우유 녹차가루 박력분 아몬드가루 녹차가루 포도씨유 설탕 소금 by 강혜</p>

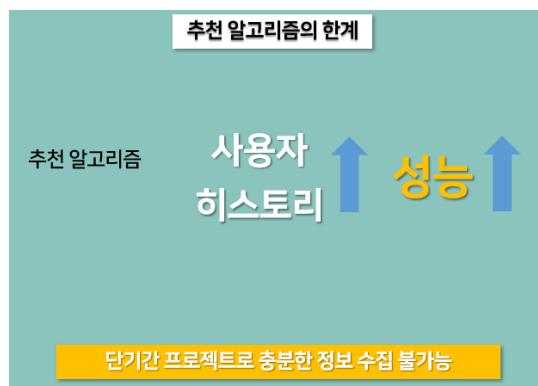
[그림 44] 사용자 추천 결과

평소 케이크 등의 디저트를 즐겨먹는 사용자의 추천 목록 결과다. 다양한 종류의 케이크뿐만 아니라 디저트로 분류될 수 있는 '쿠키', '타르트', '와플', '마카롱' 등이 추천되었으며 다른 비슷한 취향의 사용자가 '스테이크'와 '피자'를 선호했는지 추천 리스트 수록된 것을 통해 확인할 수 있었

다. 단편적인 사례만으로 추천 시스템의 성능을 평가할 수 없지만, 오차 범위가 그렇게 크지 않은 것을 팀원들 간의 수백번의 테스트와 추후에 다루고 분석할 사용자 Study를 통해 확인할 수 있었다. 따라서 위의 과정에서 볼 수 있듯이, Content-based-Filtering과 사용자들의 정보를 기반으로한 Latent Factor Collaborative Filtering을 혼합한 Hybrid Filtering방식의 모델 구현에 성공했다.

7.9. 한계점 및 발전방안

7.9.1 단기간 프로젝트로 충분한 정보 수집 불가능 및 추천 시스템의 검증 어려움



[그림 45] 추천 알고리즘의 문제

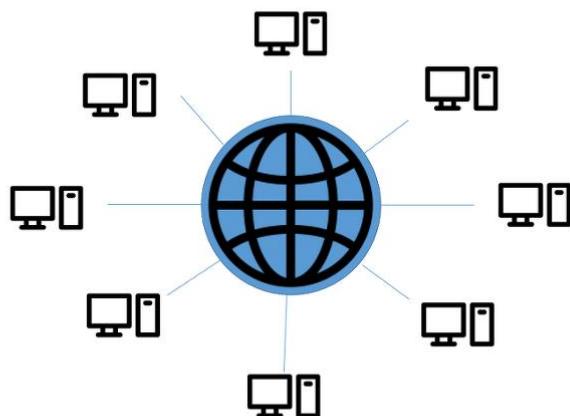
단기간 프로젝트였기에 사용자를 확보와 데이터 축적면에서 어려움이 있었다. User Study에 참여한 사용자는 15명에 불과한데, 15명 내에 비슷한 취향의 사용자들이 있는 것은 현실적으로 가능성이 희박하다. 따라서 콘텐츠 기반의 추천과 사용자 기반의 추천의 우선순위를 둘 때 아직 사용자 기반의 추천이 정확도가 우수하지 못하기 때문에 콘텐츠 기반의 추천의 우선순위를 더 높게 주었다(대략 그 비율은 8:2 ~ 9:1 이다). 훌륭한 추천 시스템을 보이는 YouTube, NetFlix 등은 방대한 콘텐츠를 가지고 있을 뿐만 아니라 수억 명의 사용자를 가지고 있기 때문에 해당 회사들의 알고리즘의 훌륭함을 뒷받침할 수 있었던 것이다. Recipe_Helper는 단기간 프로젝트였기 때문에, 만약 이 앱을 지속적으로 발전시켜 나간다면 사용량 증가로 비슷한 취향의 사용자를 더욱 찾아내서 알고리즘 성능이 개선될 것으로 예상한다.

7.9.2 무료 서버의 한계



[그림 46] 서버 성능의 한계로 계산 범위를 줄이는 과정

[그림 46]에서 볼 수 있듯이, 무료 서버를 사용했고 분산 시스템을 사용하지 않았기 때문에 레시피간 유사도를 측정하는 것에 제약이 있었다. 당초에는 모든 레시피 10만개 간의 유사도를 측정하고자 했다. 그러나 6200시간이 예상 시간으로 나오자 계산 범위를 축소하는 것으로 계획을 변경했다. 100개의 food_group을 확인한 결과 가장 낮은 레시피의 개수를 가진 group이 500개였기 때문에 사용자가 가장 많은 관심을 보인 food_group 3개를 선택하여 각 group마다 500개를 임의로 추출한 뒤 유사도를 계산했다. 그러나 이 역시 15~20초의 계산 시간이 소모되어 더욱 축소한 100개를 선택하여 계산하였고, 그 결과 최대 3초가 소모되어 100개를 임의 추출한 뒤 유사도를 계산하도록 했다.



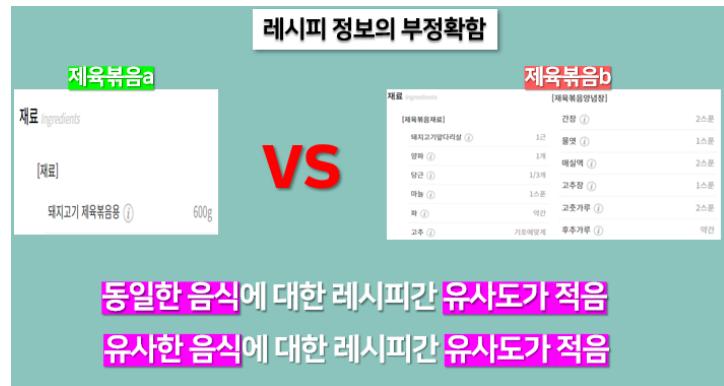
[그림 47] 분산 시스템 예시

만약 고성능의 서버를 사용하고 하둡(Hadoop)과 같은 분산 처리 시스템을 구축했다면 계산 속도의 제약으로 인한 추천 시스템의 성능 제한을 줄일 수 있었을 것이다.

7.9.3 한국어의 불규칙성, 불완전성

이는 전처리과정에서 그 이유와 처리 과정을 설명했기 때문에 생략한다.

7.9.4 레시피 정보의 부정확함



[그림 48] 레시피 정보의 부정확함

사용자마다 재료 표기의 정확도에 따른 차이가 있었다. 첫째는 동일한 요리임에도 불구하고 재료 정보가 부정확하여 유사관계가 떨어지는 레시피를 추천하게 되는 문제가, 두번째는 유사한 요리임에도 불구하고 간장, 고추장, 야채 등 부재료, 조미료 때문에 유사관계가 떨어지는 레시피를 추천하게 되는 문제가 있었다.



[그림 49] 예시: 좌측(동일한 요리의 유사도 측정 문제), 우측(유사한 요리의 유사도 측정 문제)

첫번째 문제의 예시로 다음과 같이 비슷한 제육볶음이 아닌 더덕구이와 더 유사하다는 결과가 나왔고 두번째 문제는 제육볶음이 불고기 구이가 아닌 더덕구이와 유사하다는 결과가 나왔다.

7.9.5 기존 Euclidean Distance의 문제점과 가중치 조정



[그림 50] 예시: 주재료의 가중치 조정

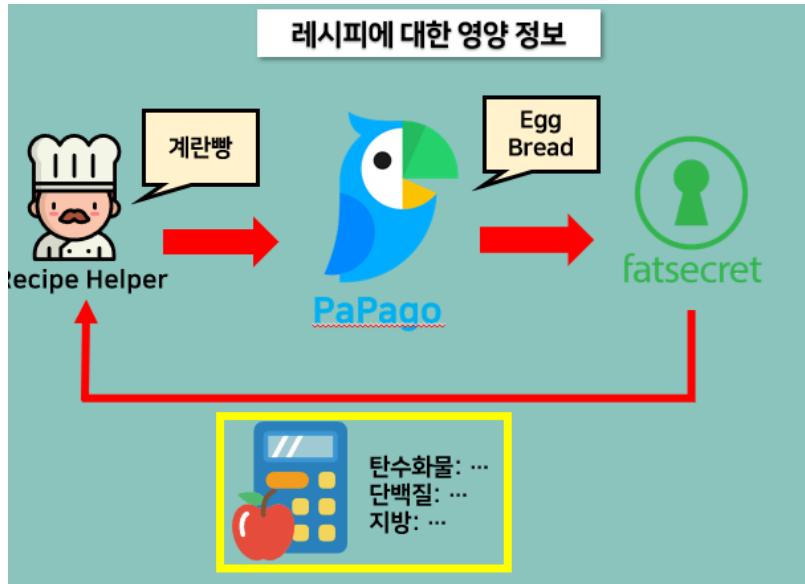
레시피 정보의 부정확함으로 인한 유사도 계산의 문제는 모든 Feature에 대해서 동일하게 계산하기 때문에 Euclidean Distance 방식의 측정 방식이 그 원인이라 파악했다. 따라서 [그림 50]과 같이 300여개의 재료 중 레시피의 정체성을 결정하는 주재료를 선정하여 가중치 조정을 했고 결과적으로 레시피간 유사도를 정확하게 측정할 수 있었다.

id	class	마늘	소금	간장	양파	설탕	대파	물	후추	...	class_93	class_94	class_95	class_96	class_97	class_98	class_99	
104950	6875705	99	0	0	0	0	1	0	5	0	...	0	0	0	0	0	0	5
104720	6941727	99	0	0	0	0	0	0	5	0	...	0	0	0	0	0	0	5
104937	6881987	99	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	5
104675	1978401	99	0	0	0	0	1	0	5	0	...	0	0	0	0	0	0	5
104867	6900448	99	0	0	0	0	0	0	5	0	...	0	0	0	0	0	0	5
104708	6946841	99	0	1	1	0	0	0	5	0	...	0	0	0	0	0	0	5
105043	6839794	99	0	0	0	1	0	0	0	0	...	0	0	0	0	0	0	5

[그림 51] 예시: Food_group 을 Feature 로 추가하여 가중치 조정한 결과

그러나 위의 조치에도 불구하고 테스트 도중 약 2~3%정도 유사도가 떨어지는 음식이 유사도가 판단되는 경우가 있었다. 따라서 [그림 51]과 같이 food_group를 Feature로 추가하여 가중치를 주재료와 같이 높게 주었고, 더 이상 유사도가 떨어지는 음식이 나타나지 않게 되었다.

7.9.6 레시피의 영양 정보



[그림 52] Recipe Helper 의 영양정보 획득 과정

식약처에서 제공하는 음식에 대한 영양 정보는 대부분 시중에 파는 완성품을 기준으로 영양 정보를 제공하고 단위가 통일되지 못했다는 점으로 사용하기에 알맞지 않았다. 따라서 레시피에 대한 영양 정보는 수 천만명의 사용자를 보유한 식단관리 어플인 Fatsecret이 제공하는 음식 영양 정보 api를 활용했다. 그러나 Fatsecret은 한글 api를 사용하기 위해서는 추가 사용료를 지불해야 했기 때문에, 영어로 이를 변환해야 했다. 따라서 네이버에서 제공하는 번역 기능인 Papago api를 사용해 문제를 해결했다. 하지만 식재료를 통해 영양정보를 계산하는 것이 아니라 요리 이름으로 영양 정보를 계산하는 것이라 오차 범위가 조금씩 있다. 하지만 음식을 구성하는 영양 성분비는 거의 일치하기 때문에 의미가 있었다고 판단했다. 위 과정을 요약하면 다음과 같다.

- 사용자가 '계란빵'a에 대한 레시피의 영양 정보를 클릭한다.
- Recipe Helper APP에서 해당 레시피의 제목을 Papago Api에 전달하면 이를 'Egg Bread'로 번역한다.
- 번역한 결과를 Fatsecret Api에 전달하면 실제로 계란빵에 대한 영양 정보를 얻을 수 있었다.

무료 api를 사용하여 영양 정보에 대해 제기될 수 있는 문제는 '과연 형용사가 포함되거나 제목이 일반적이지 않는 경우에 대한 번역, 검색 과정이 잘 이루어지는가?' 이다. 정답은 '대부분 그렇다'이다. 문제점을 2가지로 나누어서 테스트해보았다. 아래는 그 예시이다.

1. 제목이 '치즈 계란빵'과 같이 형용사가 포함된 경우
2. 제목이 '아이들 간식으로 적합한 계란빵!!'과 같이 불필요한 정보가 들어간 경우

1번 문제의 경우 'Cheese Egg Bread'로 번역이 되어 올바른 결과를 도출했다.

2번 문제의 경우는 대체로 정확했지만 잘못된 결과를 출력한 경우가 있었다. Fatsecret의 검색체계가 정확히 어떠한지 알 수 없지만 '아이들 간식으로 적합한 계란빵!!'의 경우 단어로 유의미한 것은 'snack'과 'Egg bread'인데, 'snack'을 검색하여 계란빵이 아닌 과자에 대한 영양 정보를 도출하는 경우가 있었다. 이러한 경우, 음식에 대해 미리 Tokenizing을 하여 이에 대한 리스트를 작성하여 매 검색마다 확인하는 방법이 있다. 그러나 위의 우회 방법은 무료 api를 사용하여 생긴 issue이기 때문에 만약 유료 api를 사용했다면 훨씬 더 정확한 영양 정보를 얻을 수 있었을 것이다.

7.9.7 각 레시피에 대한 Title의 문제

recipe_id	title	author	time	level	photo	class_name	ingred	class_num
0 6908683	에어프라이어/전자렌지로 만든 "모닝계란빵"	서재인	30분 이내	아무나	https://recipe1.ezmember.co.kr/cache/recipe/20...	굽기_빵	파슬리가루 계란 헹 모닝빵 마요네즈 소금 모닝빵 햄	0
1 6840149	겨울방학 아이간식 홈메이드 계란빵	eungooon	5분 이내	아무나	https://recipe1.ezmember.co.kr/cache/recipe/20...	굽기_빵	소금 계란 우유	0
2 6877579	모닝빵으로 5분만에 계란빵을 만들어보세요	gojasmin	15분 이내	아무나	https://recipe1.ezmember.co.kr/cache/recipe/20...	굽기_빵	계란 모닝빵 소금 후추 체다치즈	0
3 6923905	크리스마스 요리 간단 에어프라이어 계란빵	남남간단요리	20분 이내	조금	https://recipe1.ezmember.co.kr/cache/recipe/20...	굽기_빵	소금 계란 핫케이크가루	0
4 6900692	목우준 초간단 아이간식 소시지 계란빵	농협목우준	10분 이내	아무나	https://recipe1.ezmember.co.kr/cache/recipe/20...	굽기_빵	식빵 계란	0

[그림 53] 레시피 DB 예시

레시피의 제목은 레시피 작성자에 의해 유의미한 레시피에 대한 특징 정보가 포함되었을 가능성이 높다. [그림 53]에서 볼 수 있듯이, 레시피의 제목이 불규칙하고 의미 없는 정보가 많다. 따라서 이를 활용하기 위해서 NLP를 사용하고자 했지만 다음과 같은 결과가 나왔다.

```
1 [PAD]
2 [UNK]
3 [CLS]
4 [SEP]
5 [MASK]
6 에어
7 프라이어
8 /
9 전자
10 렌지
11 로
12 만드
13 "
14 모닝
15 계란빵
16 "♪
17 겨울
18 방학
19 아이
20 간식
21 흘
22 메이드
23 빵
24 으로
25 5
26 분
27 만
28 에
29 을
30 만드
31 어
```

[그림 54] 레시피의 제목을 Tokenizing 한 결과

[그림 54]에서 확인할 수 있듯이, 사실상 사용할 수 없다. 따라서 음식의 맛을 표현하는 여려가지 단어(특히 형용사)와 여러 음식들을 대상으로 Tokenizing list를 만들었고, 이를 통해 키워드를 추출해보았다.

```

spicy_lst = {'매운': 0, '매콤': 0, '스트레스': 0, '맵다': 0, '매콤': 0, '땀이': 0}
sour_lst = {'신맛': 0, '쓴': 0, '시큼': 0, '새콤': 0}
calcal_lst = {'얼큰': 0, '퀄콸': 0}
sweat_lst = {'달콤': 0, '달달': 0, '감미': 0, '달짝': 0, '달착': 0, '허니': 0, '꿀': 0, '스위트': 0, '스윗': 0}
saulty_lst = {'간장': 0, '짭조': 0, '짭짜': 0, '짠': 0, '짭잘': 0, '짭짤': 0, '데리야끼': 0}
soup_lst = {'국물': 0, '국': 0, '탕': 0, '감칠': 0}
etc_lst = {'따뜻': 0, '뜨끈': 0, '는든': 0,
            '찌개': 0, '간장': 0,
            '라면': 0, '냉면': 0, '쫄면': 0, '우동': 0, '스파게티': 0, '당면': 0,
            '라면': 0, '짬뽕': 0, '짜장': 0, '캘리포니아': 0, '탕수육': 0, '카레': 0, '쫄': 0,
            '초밥': 0,
            '돼지': 0,
            '제육': 0,
            '계란': 0,
            '소고기': 0, 'LA갈비': 0,
            '우유': 0, '씨리얼': 0,
            '소시지': 0, '햄': 0, '스팸': 0, '소세지': 0,
            '간편': 0, '초스피드': 0, '자취': 0, '간단': 0,
            '닭': 0,
            '구이': 0, '찜': 0, '무침': 0, '볶음': 0, '조림': 0, '부침': 0,
            '빵': 0, '브레드': 0,
            '샌드위치': 0, '마들렌': 0, '머랭': 0, '머핀': 0, '마카롱': 0,
            '브라우니': 0, '타르트': 0, '꼬치': 0, '빙수': 0, '와플': 0, '피자': 0,
            '날치알': 0,
            '호밀': 0, '통밀': 0, '바게트': 0,
            '두부': 0,
            '콩': 0,
            '호두': 0,
            '견과': 0,
            '감자': 0, '고구마': 0,
            '허니': 0, '꿀': 0,
            '쿠키': 0,
            '갈릭': 0, '마늘': 0,
            '초코': 0,
            '크림': 0,
            '버터': 0,
            '치즈': 0, '로제': 0,
            '스테이크': 0,
            '등심': 0,
}

```

[그림 55] 예시: Tag 추출을 위한 Tokenizing list

recipe_id	title	author	time	level	photo	class_name	ingred	class_num	tag
0 6908683	에어프라이어/전자렌지로 만든 "모닝계란빵"	서재인	30분 이내	아무나	https://recipe1.ezmember.co.kr/cache/recipe/20...	굽기_빵	파슬리가루 계란 햄 모닝빵 마요네즈 소금 모닝빵 햄	0	[계란, 빵]
1 6840149	겨울방학 아이간식 풀메이드 계란빵	eungooon	5분 이내	아무나	https://recipe1.ezmember.co.kr/cache/recipe/20...	굽기_빵	소금 계란 우유	0	[계란, 빵, 간식]
2 6877579	모닝빵으로 5 분만에 계란빵을 만들려보세요	gojasmin	15분 이내	아무나	https://recipe1.ezmember.co.kr/cache/recipe/20...	굽기_빵	계란 모닝빵 소금 후추 체다치즈	0	[계란, 빵]
3 6923905	크리스마스 요리 간단 에어프라이어 계란빵	남남간단 요리	20분 이내	조금	https://recipe1.ezmember.co.kr/cache/recipe/20...	굽기_빵	소금 계란 핫케이크가루	0	[계란, 간단, 빵]
4 6900692	목우준 초간단 아이간식 소시지 계란빵	농협목우준	10분 이내	아무나	https://recipe1.ezmember.co.kr/cache/recipe/20...	굽기_빵	식빵 계란	0	[계란, 소시지, 간단, 빵, 간식]

[그림 56] Tag 추출 결과 1

106865	6858235	대하회 살이 팽글탱글 해야 맛있어요~!	판교댁송 블리	30분 이내	조금	https://recipe1.ezmember.co.kr/cache/recipe/20...	회_수산물	얼음 간장 대 하 맛술	100	[회]
106866	6857073	회 뜨고 남은 생선 매운탕	조이맘	30분 이내	아무 나	https://recipe1.ezmember.co.kr/cache/recipe/20...	회_수산물	청양고추 떡 콩나물 마늘 깻잎 무 대파 미나리	100	[매 콤, 국 물, 생 선, 회]
106867	6848118	자연산 참돌회 유비끼 집에서 맛있게 즐기기	전라도언 니루비씨	15분 이내	아무 나	https://recipe1.ezmember.co.kr/cache/recipe/20...	회_수산물	고추냉이 간장 초고추장 물	100	[참 돌, 회]
106868	4757441	아삭아삭 영양만점 '죽 순회'	cooking	90분 이내	조금	https://recipe1.ezmember.co.kr/cache/recipe/20...	회_수산물	고춧가루 올리 고당 마늘 깨 고추장 식초	100	[영 양, 회]
106869	2565820	날치알 참치회 무쌈	오즈토토	30분 이내	아무 나	https://recipe1.ezmember.co.kr/cache/recipe/20...	회_수산물	참치 딸기 날 치알 파프리카 김	100	[날 치 알, 참 치, 회]

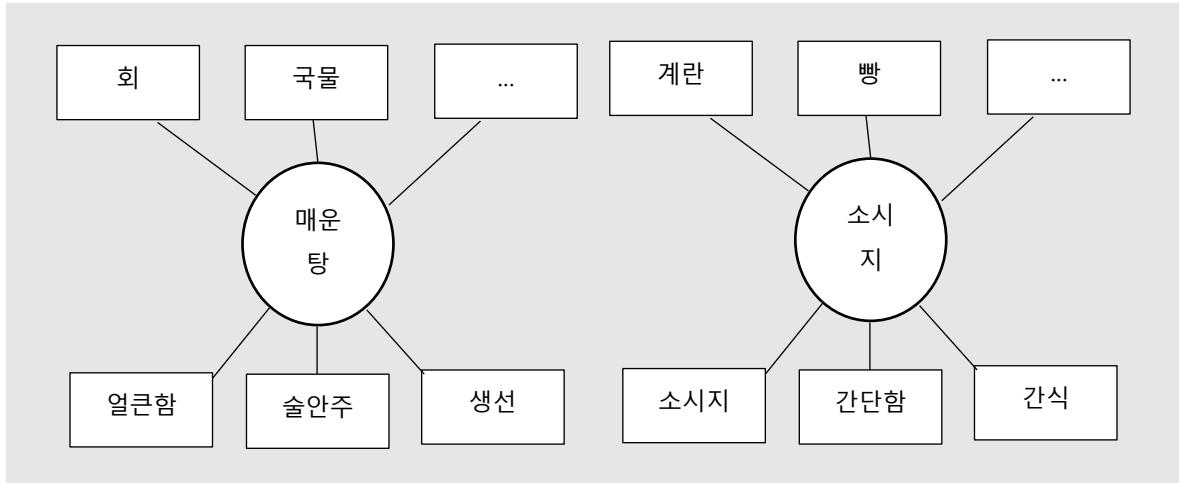
[그림 57] Tag 추출 결과 2

recipe_id		title	author	time	level	photo	class_name	ingred	class_num	tag
106770	1878270	아보카도크림스프	야야코시	30 분 이내	아무 나	https://recipe1.ezmember.co.kr/cache/recipe/20...	회_아보카도	우유 양파 생크 림 소금 아보카 도	102	[크 림, 아 보 카 도]
106771	1763315	아보카도 미니마린	야야코시	30 분 이내	조금	https://recipe1.ezmember.co.kr/cache/recipe/20...	회_아보카도	박력분 생크림 바닐라아이스 향 버터 블렌딩 베이 킹파우더 설탕 아보카도	102	[머 핀, 아 보 카 도]
106772	6876070	사케동 만드는 법 쉽고 간단한 한그릇 레시피,	먹순	10 분 이내	아무 나	https://recipe1.ezmember.co.kr/cache/recipe/20...	기타_회덮밥	간장 고추냉이 새싹채소 양파 깻잎 소금 물 마 늘 얹어 설탕 식 초	72	[간 단, 사 케 동]
106773	6870071	내맘대로 타르타르소스 곁들인 야채동魄 훈제 연어롤~	촌아짐	30 분 이내	아무 나	https://recipe1.ezmember.co.kr/cache/recipe/20...	회_밥	파프리카 오이 지 얹어 양파 마 요네즈 레몬즙 후추 파프리카 설탕 오이	48	[롤, 연 어, 훈 제]
106774	6883593	크래미 연어롤 & 허스 래디쉬소스	쑤리	60 분 이내	조금	https://recipe1.ezmember.co.kr/cache/recipe/20...	회_밥	오이 식초 양파 마요네즈 물 소 금 후추 양파 크 래미 설탕 피클 김밥김 밥	48	[롤, 연 어]

[그림 58] Tag 추출 결과 3

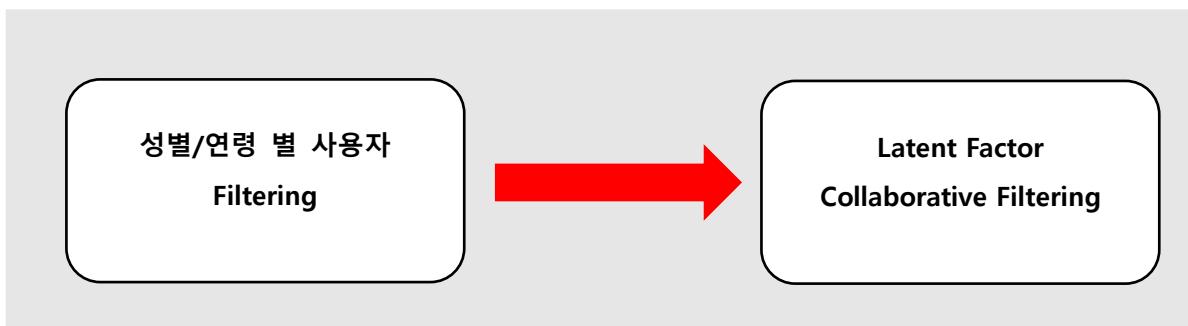
[그림 56]~[그림 58]은 Tag의 추출 결과를 보여주는 것으로 임의의 DataFrame을 추출한 결과이다. 전반적으로 Tagging이 정확하게 되었음을 확인할 수 있다. 특히, '목우촌 초간단 아이간식 소세시 계란빵'에서 특징으로 '간단함', '간식용', '소시지', '계란빵'을 추출한 것, '회 뜨고 남은 생선 매운탕'에서 '회', '생선', '매콤', '국물'을 추출한 것을 통해 레시피의 특징을 더욱 자세히 나타냈음을

볼 수 있다. 이를 활용하여 레시피의 특징뿐만 아니라 음식의 특징 역시 일반화할 수 있는 흥미로운 결과도 확인했다. 위 결과를 통해 사용자의 개인별 취향을 분석하고 추측을 더 강화할 수 있었다.



[그림 59] 예시: 음식 및 레시피에 대한 특징 추출

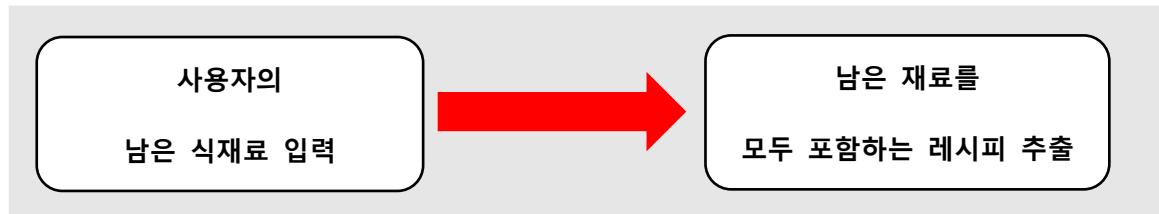
7.6 (성별/연령별) 추천 모델



[그림 60] 성별/연령별 추천 모델

사용자의 성별/연령별 정보를 기준으로 사용자 Group을 생성하여 Latent Factor Collaborative Filtering을 사용한다. 추천 시스템을 사용하기 앞서, 사용자의 개인 정보를 활용하여 필터링하는 것 외에는 차이가 없다.

7.7 식재료별 추천



[그림 61] 식재료별 추천

[그림 61]과 같이, 사용자로부터 남은 식재료를 입력 받으면 총 10만개의 레시피 중 해당 재료를 포함한 레시피들을 필터링한 결과를 사용자에게 보여준다.

The screenshot displays two side-by-side search results for different sets of ingredients.

Left Side (Ingredients: 김, 소고기):

- 나의 냉장고**: Shows the available ingredients: 김 (Kimchi) and 소고기 (Beef).
- 남아있는 식재료**: Shows the available ingredients: 김 (Kimchi), 소고기 (Beef), 돼지고기 (Pork), 양파 (Onion), 두부 (Tofu), 대파 (Garlic Chives), and 상추 (Spinach).
- Helper의 Solution 검색결과 50개**: Displays 50 search results. One result is highlighted:
 - 육우로 만든 도시락 ; 묵은
지 소고기 말이**
 - 
 - 간장 설탕 쪽파 밥 김치 소고기 물
마늘
 - by 달콤스윗맘
 - 말이 60분 이내 아무나
- 백종원 소고기튀김덮밥**
- 
- 쪽파 대파 소고기 튀김가루 소금
후추 설탕 간장 식초 맛술
- by 혀니ㅋ
- 덮밥 30분 이내 초급

Right Side (Ingredients: 돼지고기, 김, 양파, 대파, 두부, 상추):

- 나의 냉장고**: Shows the available ingredients: 돼지고기 (Pork), 김 (Kimchi), 양파 (Onion), 대파 (Garlic Chives), 두부 (Tofu), and 상추 (Spinach).
- 남아있는 식재료**: Shows the available ingredients: 돼지고기 (Pork), 김 (Kimchi), 양파 (Onion), 대파 (Garlic Chives), 두부 (Tofu), and 상추 (Spinach).
- Helper의 Solution 검색결과 32개**: Displays 32 search results. One result is highlighted:
 - 삼겹살 김밥만들기**
 - 
 - 김 밥 깨 삼장 돼지고기 깻잎 김치
참기름 고추 상추
 - by 막음입니다
 - 밥 60분 이내 초급
- 대파, 두부, 양파를 빼면 만들 수 있어요
#삼겹살볶음밥 [고추장삼겹
살볶음밥]**
- 
- 상추 깨잎 계란 김 밥 설탕 고추장
버터 김치 돼지고기 체다치즈
- by 러빙밍
- 밥 15분 이내 초급

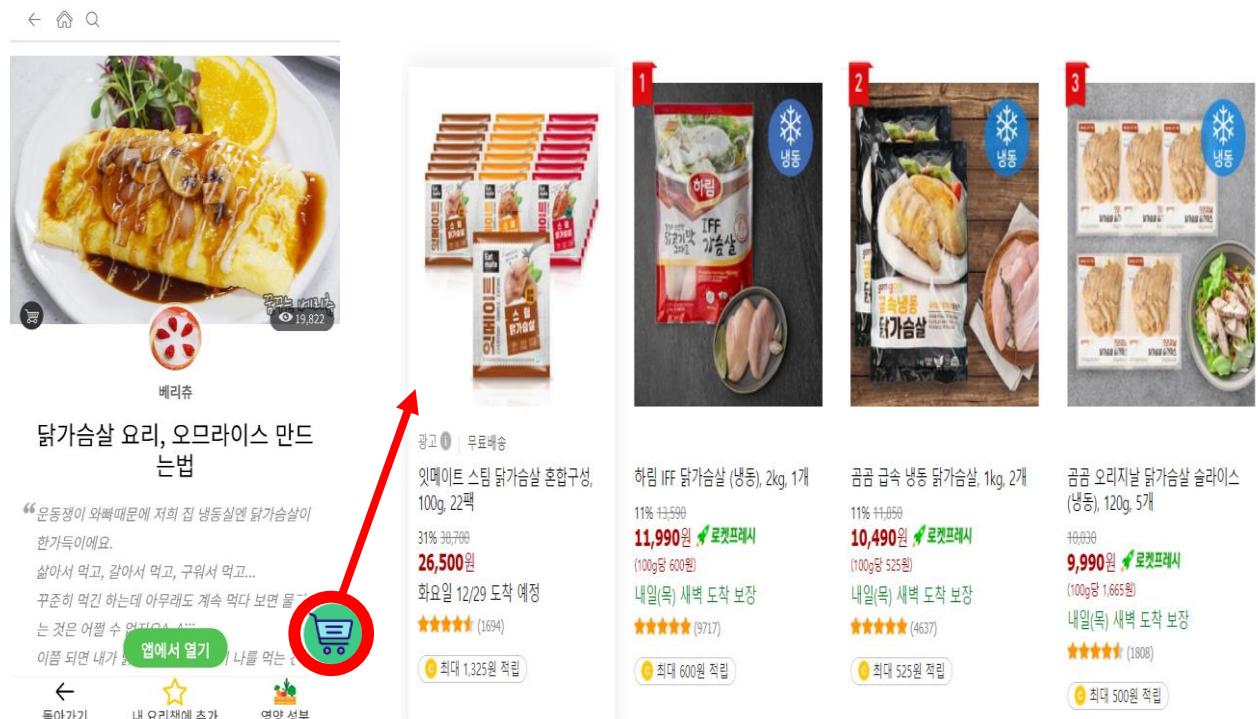
Bottom Navigation Bar:

- Home icon
- Refrigerator icon labeled "냉장고"
- Star icon
- Instagram icon
- User profile icon
- Home icon
- Refrigerator icon labeled "냉장고"
- Star icon
- Instagram icon
- User profile icon

[그림 62] 식재료 기반의 레시피 추천 결과

좌측의 경우 남아있는 재료로 만들 수 있는 레시피가 충분하지만 우측의 경우 냉장고에 등록된 식재료가 많아 이를 통해 만들 수 있는 레시피가 적을 수 있다. 따라서 결과로 10개 이하의 레시피가 나올 경우 현재 등록된 재료 중 하나씩 제거하여 10개가 넘을 때까지 레시피를 찾는다. 만약 10개를 초과한다면 탐색을 멈추고 해당 재료가 빠져야 함을 알려준다.

혹은 반대로 재료가 필요한 경우도 있을 것이다. 이러한 경우는 구현하지 않았지만, 필요한 재료를 명시하여 다음과 같이 쿠팡과 같은 E-Commerce와 연동하여 구매하도록 유도할 수도 있다.



[그림 63] Ecommerce 와의 연동으로 발전 방안

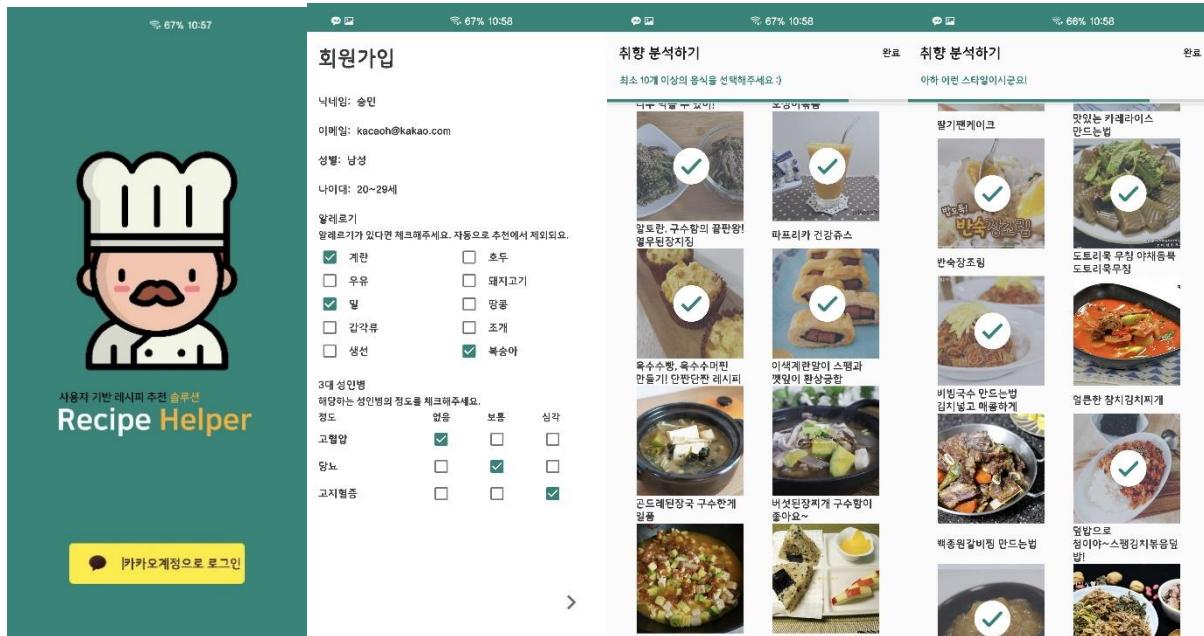
7.10. 결론

결론적으로 Content-based-Filtering과 사용자들의 정보를 기반으로한 Latent Factor Collaborative Filtering을 혼합한 Hybrid Filtering방식의 모델 구현에 성공했다. 또한 Model 개발 과정에서 부딪히는 한계를 7.9에서 모두 설명했고 이에 대해 Recipe_Helper가 어떻게 해결했는지에 대해 자세히 서술했다. 특히 대부분의 문제는 무료로 사용한 서비스에 의해서 비롯되었으며 이는 '기술 상의 불가능'이 아님이 중요했다. 또한 각 부분에서 발전시킬 수 있는 방안에 대해 서술했다. 특히 사용자가 직접 작성한 글에서 특징을 추출하여 레시피의 재료뿐만 아니라 제목을 통해서도 레시피의 특성을 모델링할 수 있었다.

8. Implementation Result

'Recipe Helper'는 사용자 스터디를 진행하기 전 완벽하게 작동하는 어플리케이션을 구현하기 위해 노력하였다. 그 결과 기존에 기획하였던 대부분의 기능들을 모두 구현 완료하였다. 각각의 기능은 아래에서 자세하게 설명한다.

8.1. 회원가입



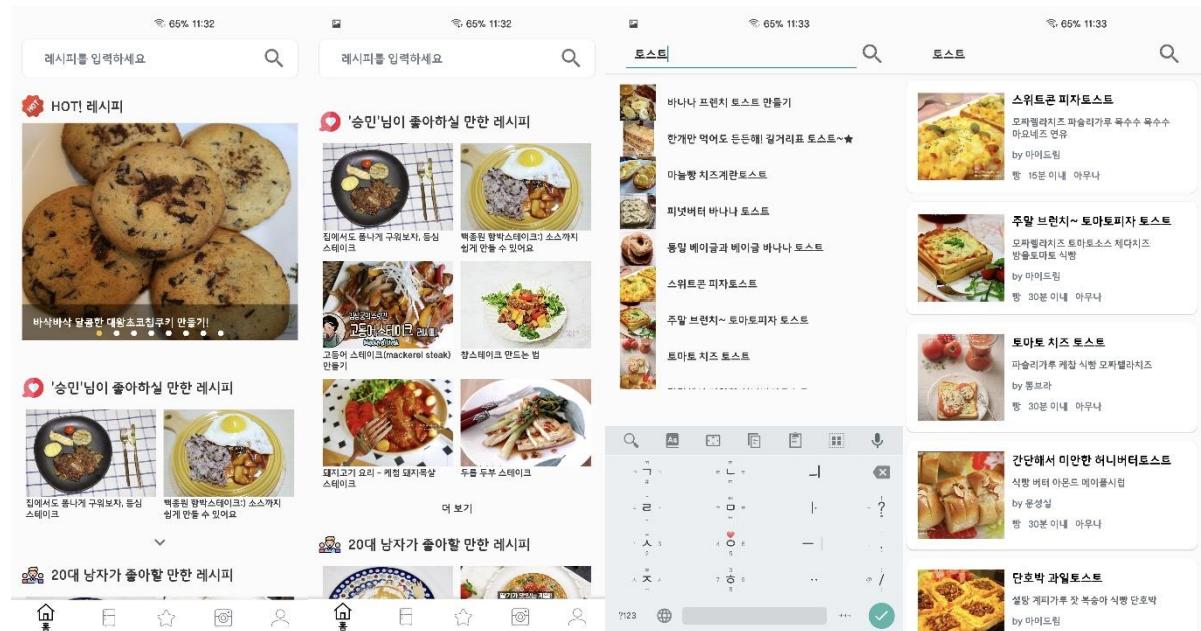
[그림 64] 카카오 로그인, 회원가입, 취향분석 1, 취향분석 2

- ✓ 카카오 로그인을 통한 간편한 회원가입
- ✓ 쉽게 자신의 건강상태 설정
- ✓ 초기 취향 분석으로 콜드 스타트 문제 방지
- ✓ Progress bar를 통해 사용자에게 현재 상황을 인지
- ✓ 재치 있는 comment로 약간의 즐거움 선사

회원가입은 쉬운 접근성을 위해서 국내 스마트폰 사용자가 대부분 사용하는 '카카오 로그인'을 통해 가입할 수 있고, 자신의 알레르기 유무와 건강상태를 Check Box 형태로 입력할 수 있다. 회원 가입 초기에는 사용자에 대한 정보가 부족해 제대로 된 추천을 해줄 수 없는 콜드 스타트 문

제에 부딪힐 수 있는데, 이런 콜드 스타트 문제를 줄이고자 기초적인 취향 분석을 진행한다. 취향 분석은 최소 10개~최대 100개까지 자신이 좋아하는 레시피를 선택할 수 있고 Progress bar를 통해 현재의 진행 사항과 가볍게 즐길 수 있는 멘트를 보여준다.

8.2. 흄

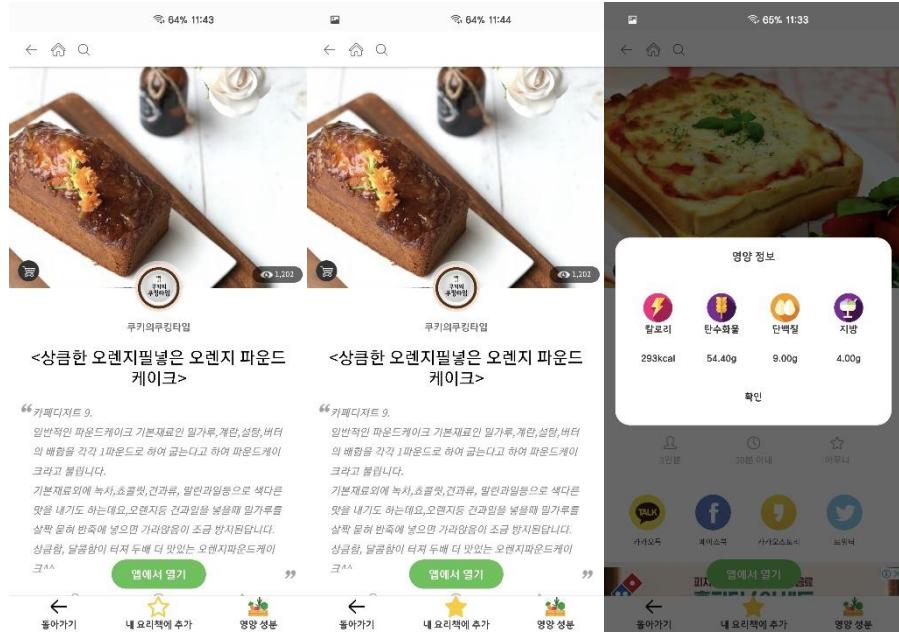


[그림 65] 흄, 펼쳤을 때, 검색할 때, 검색 결과

- ✓ HOT! 레시피, 사용자 맞춤 레시피, 비슷한 연령대, 성별이 좋아하는 레시피 추천
- ✓ 쉬운 검색 화면 진입
- ✓ 검색 화면 진입 시 자동 소프트키 생성 및 추천 검색어로 사용성 UP

홈 화면은 'Recipe Helper' 사용자들에게 인기있는 HOT! 레시피와 자신의 취향에 맞는 레시피, 비슷한 나이의 같은 성별의 사람들이 좋아하는 레시피를 추천할 수 있다. 어떤 형태의 레시피 추천을 받는지 한 번에 보기 위해 펼치는 방식으로 더 많은 추천 레시피를 확인할 수 있다. 추가로 홈 화면에서 바로 검색을 할 수 있고, 사용 편의성을 높이기 위해 검색창을 진입하면 바로 소프트키가 올라오고 유사 검색어도 추천해준다.

8.3. 레시피 페이지

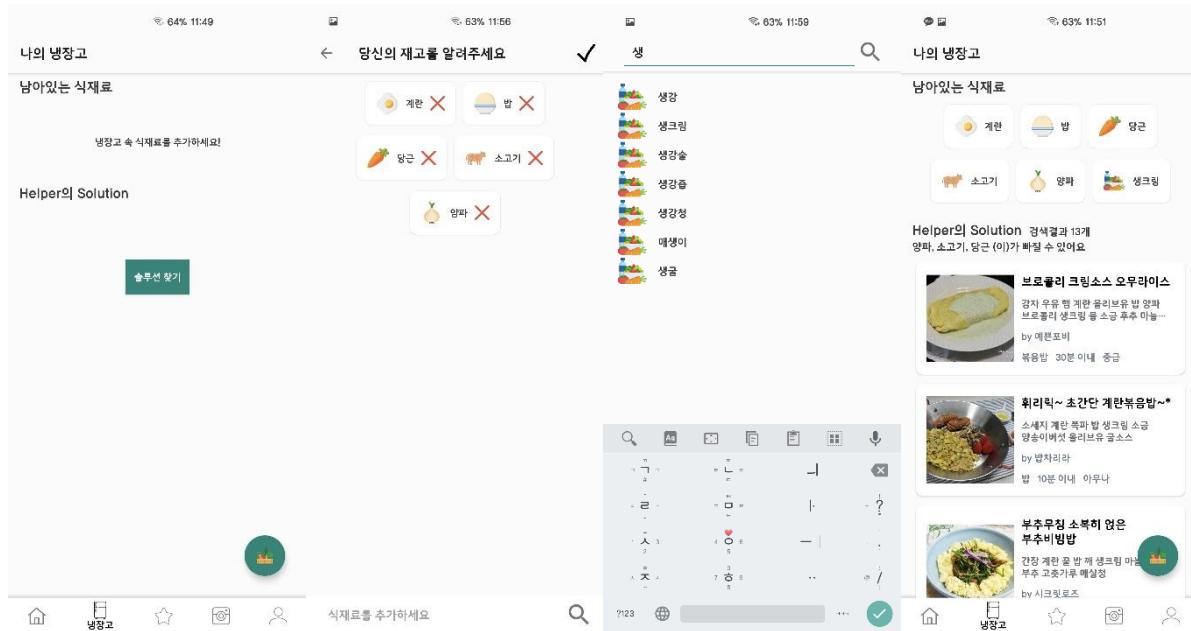


[그림 66] 레시피 페이지, 요리책 추가, 영양정보 확인

- ✓ Web View 형태로 '만개의 레시피'에 연결
- ✓ '나의 요리책'에 레시피 저장
- ✓ 영양 정보 확인

레시피 페이지는 앱 내에서 레시피를 선택했을 때 진입하는 특정 화면이다. '만개의 레시피'를 Web View 형태로 연결해주며, 페이지 내에서 해당 레시피의 영양 정보를 확인할 수 있고, '나의 요리책'에 레시피를 저장할 수 있다.

8.4. 나의 냉장고

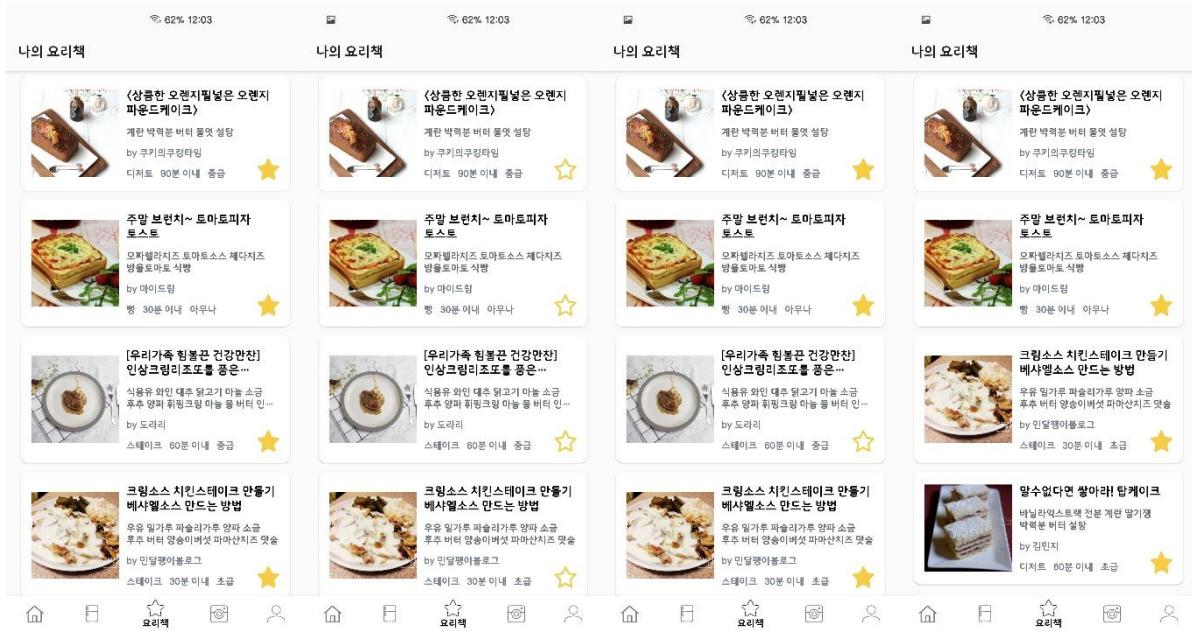


[그림 67] 냉장고 화면, 식재료 수정 화면, 식재료 추가, 레시피 검색

- ✓ 식재료 기반의 레시피 추천
- ✓ 쉬운 식재료 추가 및 삭제
- ✓ 식재료 추가 시 자동 소프트키 생성 및 추천 검색어 제공
- ✓ 레시피 추천 시 세련된 형태의 Progress bar 제공

나의 냉장고 기능은 남아 있는 식재료로 레시피를 추천하는 기능이다. 오른쪽 하단의 버튼을 통해 쉽게 남아 있는 식재료를 추가, 삭제할 수 있으며 마찬가지로 식재료를 추가할 때에는 자동으로 소프트키를 생성해주고 추천 검색어를 제공한다. 또한, 상위 100개의 식재료에 대해서는 아이콘을 함께 제공해 가시성을 높였다. '레시피 검색' 버튼을 통해 식재료 기반의 레시피 추천을 받을 수 있고, 만약 레시피 추천에 빠지는 재료, 즉 사용하지 않는 재료가 있다면 알려준다. 식재료 기반의 레시피 추천은 추천에 약간의 시간이 걸리므로 사용자에게 애플리케이션이 잘 작동하고 있음을 알려주기 위해 반짝이는 형태의 Progress bar를 제공한다.

8.5. 나의 요리책

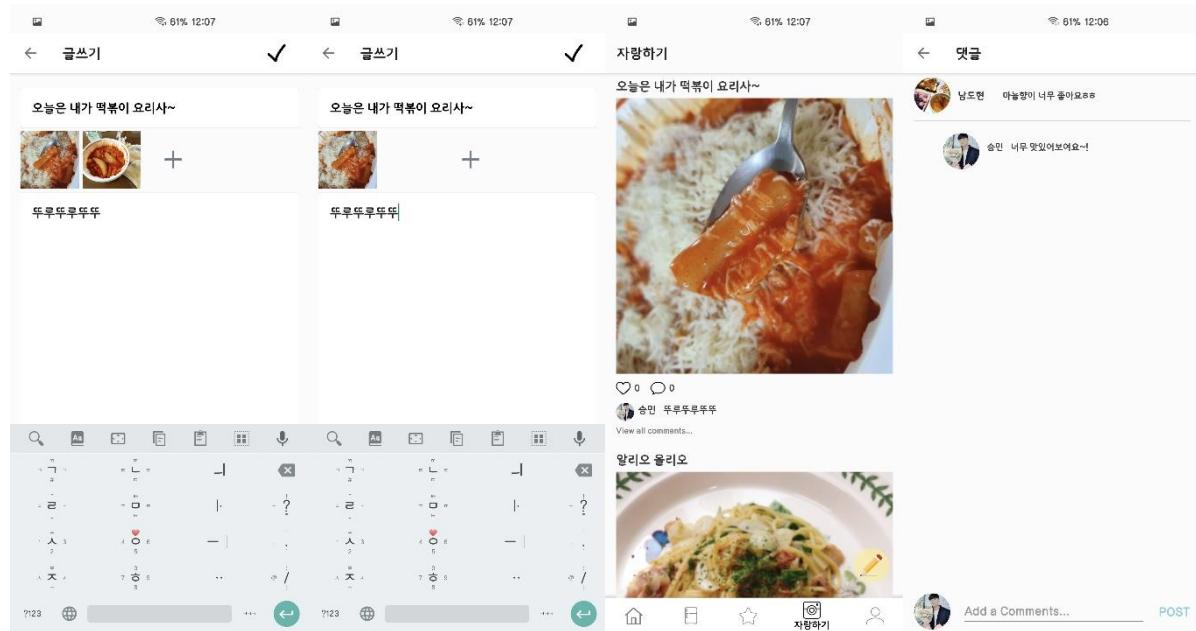


[그림 68] 요리책 화면, 스크랩 해제, 일부 스크랩 다시 선택, 반영 결과

- ✓ 레시피 저장 페이지
- ✓ 실수로 스크랩 해제하는 경우 방지로 사용성 UP

나의 요리책은 우리가 흔히 알고 있는 스크랩 페이지이다. 사용자가 스크랩한 목록을 볼 수 있으며, 별 표시를 눌러 스크랩을 해제할 수 있다. 다만, 실수로 별을 누를 경우도 많을 것으로 판단되어서 별 표시를 눌러도 요리책 목록에서 바로 사라지지 않아서 다시 눌러 스크랩을 할 수 있다. 아래 탭 버튼을 눌러서 다른 화면을 다녀오면 최종적으로 요리책 목록이 새롭게 업데이트된다.

8.6. 자랑하기



[그림 69] 글쓰기 사진 2 개 선택, 사진 1 개 삭제, 자랑하기 화면, 댓글 남기기

- ✓ 'Recipe Helper'만의 SNS
- ✓ 글쓰기 시, 선택한 사진이 무엇인지 화면에 보여줌으로써 편의성 UP
- ✓ 상황에 맞게 유연한 소프트키 생성으로 사용성 UP

자랑하기는 일종의 작은 커뮤니티이다. 자신의 요리하거나 먹은 음식 사진을 올려 다른 사용자들과 소통할 수 있다. 기본적으로 글을 작성하고 댓글을 남길 수 있다. 글을 작성할 때에는 자신이 선택한 사진이 무엇인지 알기 쉽게 보여주며 지우고 싶은 사진이 있다면 해당 사진을 단순히 누름으로써 업로드 목록에서 삭제할 수 있다. 글쓰기 화면에 진입하면 자동으로 소프트키가 생성되며, 댓글 화면에 진입할 때에는 단순히 댓글을 보기만 하려는 사용자도 많을 것으로 판단하여 자동으로 소프트키가 생성되지 않도록 하였다.

8.7. 마이페이지



[그림 70] 마이페이지, 건강상태 및 알레르기 상태 변경, 워드클라우드 변경, 설문조사 링크

- ✓ 내 정보 및 건강상태를 한 눈에 알아보기 쉽도록 구성
- ✓ 자신의 취향을 나타내는 Word Cloud 제공

마이페이지에서는 사용자가 자신의 정보들을 확인할 수 있다. 건강상태는 이모티콘과 색상을 함께 넣어서 구별하기 쉽도록 하였고, 터치를 통해 간편하게 상태를 바꿀 수 있다. 또한, 알레르기 상태도 쉽게 바꿀 수 있다. 더하여, 사용자 취향에 맞게 word cloud를 제공하는데 화면을 아래로 잡아당기면 워드 클라우드를 새로 업데이트 할 수 있다.

설문조사 링크를 마이페이지 하단에 위치해서 사용자가 쉽게 설문조사에 참여할 수 있도록 유도하였다.

9. User Study

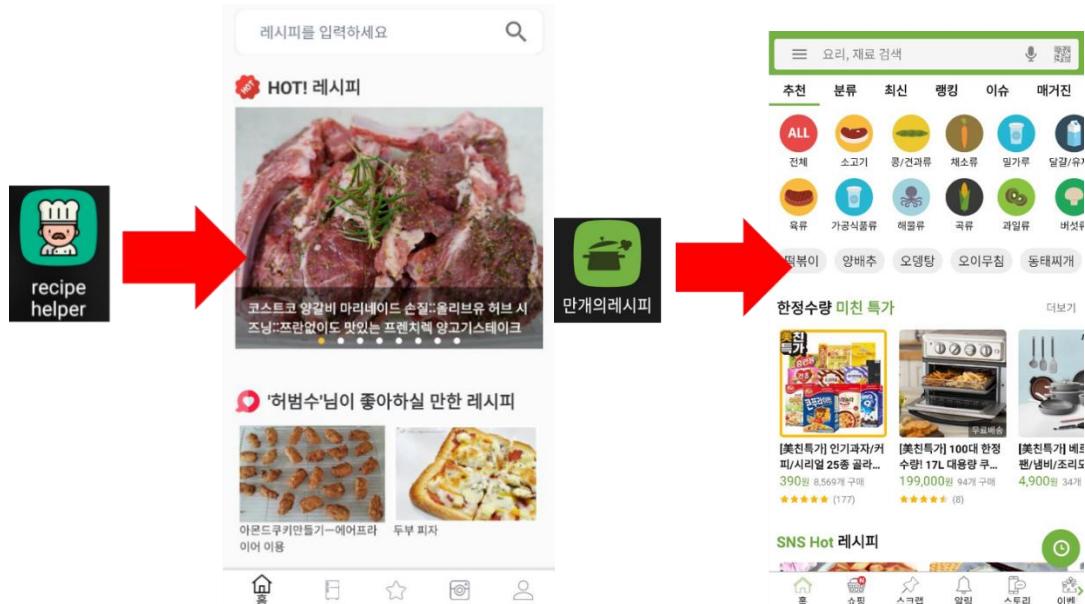
9.1. 목적

사용자 스터디를 통해 비슷한 서비스를 제공하는 시스템과 'Recipe Helper'를 비교하여 경쟁사보다 우리가 차별화된 서비스를 제공하는지 판단하는 것이 주요 목적이다. 더불어 사용자가 중요하게 생각하는 시스템 요소를 파악하여 향후 시스템 개발과 관리에 반영하고, 서비스의 사용 용이성과 기능의 목적성에 대한 만족도 등을 평가하기 위한 목적 역시 존재한다.

9.2. 데이터 수집 및 분석 계획

본 사용자 스터디에서는 비슷한 서비스를 제공하는 '만개의 레시피'와 'Recipe Helper'를 비교할 것이다.

9.2.1. 두 가지 서비스의 비교



[그림 71] Recipe Helper main(좌), 만개의 레시피 메인화면(우)

'Recipe Helper'는 기존의 서비스인 '만개의 레시피'에는 없는 기능들을 추가한 애플리케이션이다. 예를 들어서, 식재료 기반의 레시피 추천과 사용자 기반 레시피 추천, 비슷한 성향을 가진 사람의 레시피 추천 등 다양한 형태의 레시피들을 추천할 수 있는 기능들은 기존의 '만개의 레시피'에는 없던 새로운 기능이다. 따라서 두 시스템 간 만족도가 유의미하게 차이가 나는지 판별할 수 있어야한다. 그렇기 때문에 귀무 가설(H0)은 "두 서비스의 만족도를 결정하는 부분 질문들의 점수 평균은 같다."로 정하고 사용자 스터디를 진행하였다.

9.2.2. 사용자 스터디 설계

사용자 스터디 디자인 방식은 Between Subject Design으로 각 서비스당 25명씩 총 50명을 대상으로 진행한다. 사용자 sample의 수가 절대적으로 적음에도 불구하고 Within Subject Design이 아닌 Between Subject Design을 고집하는 이유는, 첫째, 'Recipe Helper'는 애플리케이션이기 때문에 시작부터 웹과 달리 사용자의 디바이스에 설치를 해야 한다는 번거로움이 존재한다. 만약 Within Subject Design으로 진행한다면, 두 개의 애플리케이션을 설치하여 사용자가 테스트를 진행해야해서 사용자가 번거로움을 느낄 수 있고, 사용자 스터디의 참여도를 떨어뜨릴지도 모르는 위험성을 가지고 있다. 두 번째 이유는 설문 조사 기간이 짧기 때문이다. 현실적으로 레시피를 보고 요리하는 행위 조차도 짧은 user study기간 동안에 사용량이 많지 않을 것으로 예상되는데(최대 7~8회 예상), 이 횟수를 절반으로 나누는 것은 사용자가 정확한 평가를 내릴 수 없을 것이라 짐작한다. 세 번째 이유는 사용자 스터디를 진행하려는 두 가지 애플리케이션 모두 인지도가 낮다는 점이다. 제일 대중적인 레시피 모음 앱이라 생각되는 '만개의 레시피'조차 인지도가 크지 않았고 특히 타겟 사용층 중 한 집단인 4~50대 이상 사용층은 젊은 10~20대 보다 스마트폰 숙련도가 떨어지기 때문에, 이들이 두 개의 애플리케이션의 동작방법을 전부 학습하는 것은 비용과 시간 양 측면에서 모두 비효율적이라고 생각했다. 이러한 근거 하에, 우리의 user study는 Between Subject Design을 토대로 진행해야 한다고 판단했다.

사용자 스터디 참여자는 두 집단으로 나뉜다. 한 집단은 본 프로젝트에서 개발한 Recipe Helper만을 사용하고, 나머지 한 집단은 비교 대상인 만개의 레시피만을 사용하게 한다. 비교 대상인 만개의 레시피는 현재 google play store에서 다운 받을 수 있고 Recipe Helper는 경우, apk 파일로 배포하였다. apk파일은 Android JAVA 환경에서 개발되었기 때문에 스터디 참여자는 안드로이드 OS기반의 스마트폰 사용자만으로 제한된다. 또한 주 사용자로 예상되는 집단인 2~30대와 40대 이상으로 사용자를, confounds를 피하기 위해 동일한 비율로 모집할 것이다. 참여자 모집에 있어 상술한 것 이외의 제약사항은 없다.

사용자 스터디 진행은 코로나 사태의 심각성을 고려해서 온라인으로 진행한다. 따라서 대면으로 진행되는 것보다 훨씬 더 자세하게 설명해야 사용자가 어려움 없이 쉽게 test를 진행할 수 있다. 다수의 사용자에게 1:1로 시간을 잡아 설명하는 것은 시간적인 제약이 있기 때문에 설치방법, 주요 기능 동작 방법의 설명서, 동영상, 혹은 필요시 전화 등을 통해 설명하는 방법을 생각했. Recipe Helper의 경우 play store에서 다운 받을 수 없기 때문에 google drive 링크를 통해 사용자가 애플리케이션을 쉽게 다운로드 받을 수 있도록 제공한다. 또한 적어도 하루에서 이를 이상 사용 후 설문을 하도록 권장하였고, 정확한 비교를 위해 두 개의 설문지는 같은 질문들로 구성하였다.

9.2.3. 사용자 스터디 문항

- 연령대가 어떻게 되시나요?
- 평소에 요리하는 빈도가 어떻게 되시나요?
- 평소에 남은 식재료 처리가 곤란한 경험이 있었나요?
- 평소에 다양한 레시피 중 하나를 선택할 때 어려움이 있었나요?

사용자 파악

- 이 애플리케이션을 사용하기 전에 기대했던 기능들이 충분히 들어가 있었나요?
- 이 애플리케이션이 본인이 생각한 목적 달성에 도움이 되었나요?

기능의 목적성

- 최종적으로 레시피를 선택하는데 얼마나 많은 시간이 걸리나요?
- 레시피에 대한 다른 사용자의 평가가 레시피 선택에 도움이 되었나요?
- 다양한 재료를 활용할 수 있도록 레시피를 제공했나요?
- 레시피에 대한 정보가 상세하게 제공되었나요? (조리/영양/재료)
- 이 어플리케이션이 실제 요리 활동 시 레시피 활용에 도움이 되었나요?

서비스 효율성

- 애플리케이션 사용 중 불편함(서비스의 지역, 오류 등)이 없었나요?
- 애플리케이션의 기능을 사용하는 과정이 복잡했나요?

사용자 용이성

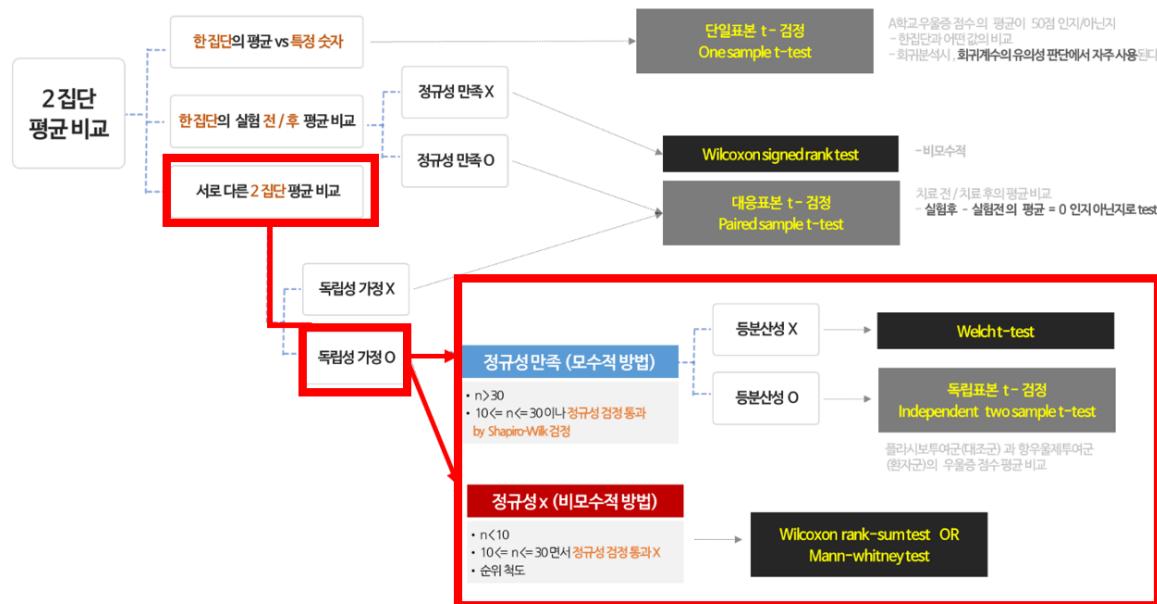
- 필요한 기능을 직관적으로, 쉽게 찾을 수 있었나요?
- 애플리케이션의 색상, 구성 등의 디자인 요소가 마음에 드셨나요?

디자인 요소

[그림 72] 사용자 스터디 파악 단위별 문항

문항은 위와 같이 '사용자 파악', '기능의 목적성', '서비스의 효율성', '사용자 용이성', '디자인 요소'를 파악하기 위한 의도로 구성되어 있다. 첫째, '사용자 파악'에서는 사용자의 연령대에 따른 레시피 검색의 사용 빈도를 파악한다. 또한 '남은 식재료'와 '선택의 어려움'이라는 질문을 통해 실제로 추천 시스템이 없는 '만개의 레시피'만을 사용한 사람들이 이러한 어려움이 겪었으며 문제로 생각하는지를 파악하고, 'Recipe Helper' 사용자에게는 실제로 제공된 기능들이 이전의 경험에 비추어서 효과가 있었는지 확인하고자 하는 목적으로 질문을 배치했다. 둘째, '기능의 목적성'을 통해 사용자들이 생각하기에 '레시피 검색' 애플리케이션이라면 가져야 할 기능들을 담고 있는지 확인한다. 셋째, '서비스의 효율성'에서는 최종적으로 Recipe Helper에서 기존의 시스템을 보완한 '사용자 기반의 추천'과 '냉장고의 재료를 사용하여 만들 수 있는 레시피에 대한 제공', '커뮤니티 기능을 통한 사용자간 레시피 공유', '레시피에 대한 영양 정보 제공 등' 이 실제로 사용자에게 있어 만족도에 차이를 주었는지 확인한다. '사용자 용이성'에서는 애플리케이션의 구현상의 오류가 있었는지, 기능을 사용자 중심으로 고려하지 못했는지 등을 파악한다. 마지막으로 '디자인 요소'를 통해 애플리케이션의 기능 배치와 디자인이 사용자에게 피로감을 주지 않고 편리성을 증대하며 감성적인 부분을 충족시켰는지 확인한다.

9.2.4. 검정 방법



[그림 73] 검정 절차

정확한 결론 도출을 위하여는 검정이 필수적이다. 보통 두 개의 집단 평균을 비교할 때는 정규성과 등분산성을 만족하느냐에 따라 검정 방법이 달라진다. [그림 73]에서 빨간 상자로 되어 있는 부분이 TomTom VIP팀이 이번에 진행할 검정 절차이다. 먼저 질문은 독립성을 만족하도록 설계하였기 때문에 만족한다. 이후 정규성을 만족하는지 판별하고 등분산성까지 판정이 이루어져야 한다.

정규성 만족을 판별하기 위해 표본의 수가 많을 때 사용하는 Kolmogorov-Smirnov(콜모고로프-스미노프) 검정과 표본의 수가 적을 때 사용하는 Shapiro Wilk(샤피로 월크) 검정이 있는데, 본 프로젝트에서는 사용자 스터디의 표본수가 많지 않기 때문에 샤피로 월크 검정으로 정규성을 검정하였다. 정규성을 검정할 때 귀무 가설(H₀)은 "두 집단은 정규분포를 따른다"로 설정하였다. '만개의 레시피'와 'Recipe Helper'의 설문 문항에 대해 두 가지 모두 가설을 수용할 경우 정규성을 만족한다고 할 수 있고 이럴 경우에는 등분산성 검정을 시행한다. 반면에 정규성을 만족하지 못할 경우 등분산성 검정을 할 필요 없이 비모수적 방식인 Wilcoxon Rank Sum 검정을 통해 해당 질문에서 설문 결과가 유의미한 차이가 있는지 확인한다. 현재 TomTom VIP팀은 p값이 0.05

등분산성 만족을 판별하기 위해 표본이 정규성을 만족할 때 사용하는 Bartlett(바틀렛) 방식과 정규성과 관계없이 판정할 수 있는 Levene(레빈) 방식이 있는데, 정규성과 상관없이 모든 문항에 대해 등분산성을 판정해보기 위해 레빈 방식을 채택하였다. 등분산성을 검정할 때 귀무 가설(H₀)은 "두 집단의 분산은 같다"로 설정하였다. '만개의 레시피'와 'Recipe Helper'의 설문 문항에 대해 두 가지 모두 가설을 수용할 경우 등분산성을 만족한다고 할 수 있고, 이 경우 Independent T 검정을 시행한다. 반면, 등분산성을 만족하지 못할 경우 자유도를 수정하는 Welch T 검정을 시행한다.

9.3. 분석 결과

모든 질문에 대해 검증을 마쳐야 하기 때문에 정규성 검증과 등분산성 검증, T 검증 코드를 작성하여 검증을 하였다.

9.3.1. 각 서비스에 대한 만족도

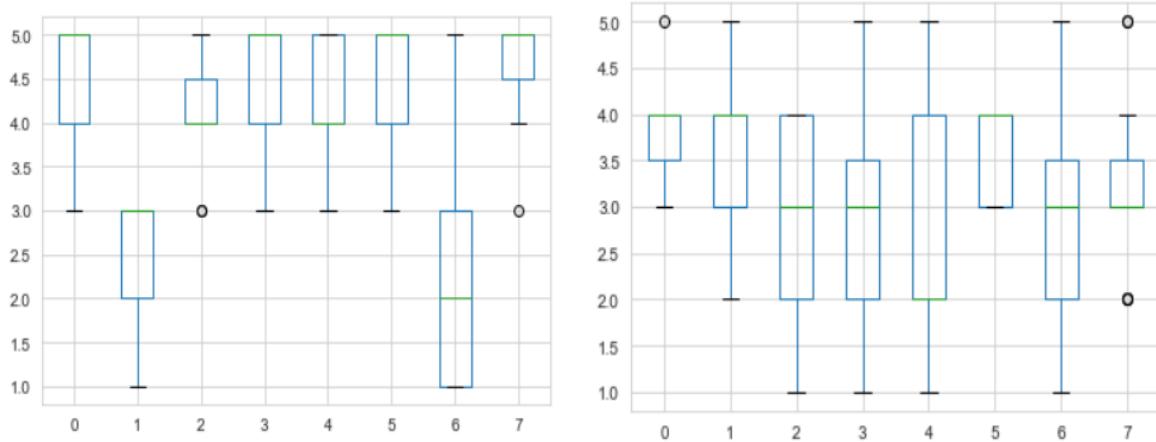
'Recipe Helper'와 '만개의 레시피'의 설문 통계 결과와 Box Plot 결과는 다음과 같다. Box Plot의 결과를 통해 대개 정규성을 따르는 것을 짐작할 수 있다. 또한 Recipe Helper가 전반적으로 만족도가 높게 나타나는 것 역시 Box plot으로 간단히 확인 가능하다.

	count	mean	std	min	25%	50%	75%	max
이 어플리케이션이 본인이 생각한 목적 달성에 도움이 되었나요?	15.0	4.400000	0.828079	3.0	4.0	5.0	5.0	5.0
최종적으로 레시피를 선택하는 데 얼마나 많은 시간이 걸리나요?	15.0	2.333333	0.816497	1.0	2.0	3.0	3.0	3.0
게시판에 있는 레시피에 대한 다른 사용자의 평가가 레시피 선택에 도움이 되었나요?	15.0	4.133333	0.639940	3.0	4.0	4.0	4.5	5.0
다양한 재료를 활용할 수 있도록 레시피를 제공했나요?	15.0	4.466667	0.743223	3.0	4.0	5.0	5.0	5.0
조리법, 영양 성분, 필요한 재료 등의 레시피에 대한 정보가 상세하게 제공되었나요?	15.0	4.266667	0.798809	3.0	4.0	4.0	5.0	5.0
이 어플리케이션을 활용해 고른 레시피가 실제로 요리할 때 도움이 됩니까?	15.0	4.400000	0.736788	3.0	4.0	5.0	5.0	5.0
어플리케이션의 기능을 사용하는 과정이 복잡했나요?	15.0	2.200000	1.373213	1.0	1.0	2.0	3.0	5.0
필요한 기능을 직관적으로, 쉽게 찾을 수 있었나요?	15.0	4.666667	0.617213	3.0	4.5	5.0	5.0	5.0

[그림 74] 만족도 통계 – Recipe Helper

	count	mean	std	min	25%	50%	75%	max
이 어플리케이션이 본인이 생각한 목적 달성에 도움이 되었나요?	15.0	3.800000	0.560612	3.0	3.5	4.0	4.0	5.0
최종적으로 레시피를 선택하는 데 얼마나 많은 시간이 걸리나요?	15.0	3.800000	0.861892	2.0	3.0	4.0	4.0	5.0
게시판에 있는 레시피에 대한 다른 사용자의 평가가 레시피 선택에 도움이 되었나요?	15.0	2.733333	1.099784	1.0	2.0	3.0	4.0	4.0
다양한 재료를 활용할 수 있도록 레시피를 제공했나요?	15.0	2.733333	1.162919	1.0	2.0	3.0	3.5	5.0
조리법, 영양 성분, 필요한 재료 등의 레시피에 대한 정보가 상세하게 제공되었나요?	15.0	2.866667	1.245946	1.0	2.0	2.0	4.0	5.0
이 어플리케이션을 활용해 고른 레시피가 실제로 요리할 때 도움이 됩니까?	15.0	3.600000	0.507093	3.0	3.0	4.0	4.0	4.0
어플리케이션의 기능을 사용하는 과정이 복잡했나요?	15.0	2.866667	1.125463	1.0	2.0	3.0	3.5	5.0
필요한 기능을 직관적으로, 쉽게 찾을 수 있었나요?	15.0	3.200000	0.941124	2.0	3.0	3.0	3.5	5.0

[그림 75] 만족도 통계 – 만개의 레시피



[그림 76] Box Plot 결과 (좌) Recipe Helper (우) 만개의 레시피

9.3.2. 정규성 검정 결과(Shapiro-Wilk)

정규성 판정 결과 대부분의 문항에서 p-value가 TomTom VIP팀이 설정한 값 0.05보다 크게 나와 정규성을 만족하였다.

```
def calShapiro(helper, thousand, alpha = 0.05):
    # Calculate helper data shapiro
    stat, p = shapiro(helper)
    print(f'HELPER statistics={stat:.3f}, p={p:.5f}', end=' ')
    if p > alpha:
        pass
        print(f" >> looks Gausian (fail to reject H0)")
    else:
        print(f" >> does not look Gausian (reject H0)")
        print("Unsatisfied with normality, Go to Wilcoxon Rank Sum Test")
        return False

    # Calculate thousand data shapiro
    stat, p = shapiro(thousand)
    print(f'THOUSAND statistics={stat:.3f}, p={p:.5f}', end=' ')
    if p > alpha:
        pass
        print(f" >> looks Gausian (fail to reject H0)")
    else:
        print(f" >> does not look Gausian (reject H0)")
        print("Unsatisfied with normality, Go to Wilcoxon Rank Sum Test\n")
        return False

    print("Satisfied with normality.\n")
    return True
```

#정규성 검정

for num in ques_num:
 print(f'{ques[num]}')
 helper_value = helper_df.iloc[:,num].values
 thousand_value = thousand_df.iloc[:,num].values
 calShapiro(helper_value, thousand_value)

Q6: 이 어플리케이션이 본인이 생각한 목적 달성을 도움이 되었나요?
 HELPER statistics=3.148, p=0.20723 >> looks Gausian (fail to reject H0)
 THOUSAND statistics=0.360, p=0.83517 >> looks Gausian (fail to reject H0)
 Satisfied with normality.

Q7: 최종적으로 레시피를 선택하는 데 얼마나 많은 시간이 걸리나요?
 HELPER statistics=2.823, p=0.24380 >> looks Gausian (fail to reject H0)
 THOUSAND statistics=0.392, p=0.82199 >> looks Gausian (fail to reject H0)
 Satisfied with normality.

Q8: 게시판에 있는 레시피에 대한 다른 사용자의 평가가 레시피 선택에 도움이 되었나요?
 HELPER statistics=0.046, p=0.97747 >> looks Gausian (fail to reject H0)
 THOUSAND statistics=2.719, p=0.25684 >> looks Gausian (fail to reject H0)
 Satisfied with normality.

Q9: 다양한 재료를 활용할 수 있도록 레시피를 제공했나요?
 HELPER statistics=3.407, p=0.18206 >> looks Gausian (fail to reject H0)
 THOUSAND statistics=0.406, p=0.81620 >> looks Gausian (fail to reject H0)
 Satisfied with normality.

[그림 77] 정규성 검정을 위한 코드와 결과

9.3.3. 등분산성 검정 결과(Levene)

등분산성 판정 결과 대부분의 문항에서 p-value가 TomTom VIP팀이 설정한 값 0.05보다 크게 나와 등분산성을 만족하였다.

```
#등분산성 검정
for num in ques_num:
    print(f'Q({num}): {ques[num]}')
    helper_value = helper_df.iloc[:,num].values
    thousand_value = thousand_df.iloc[:,num].values

    callLevene(helper_value,thousand_value)

06: 이 어플리케이션이 본인이 생각한 목적 달성을 도움이 되었나요?
LEVENE statistics=1.155, p=0.29175 >> looks Equality (fail to reject H0)
Satisfied with homogeneity of variance.

07: 최종적으로 레시피를 선택하는 데 얼마나 많은 시간이 걸리나요?
LEVENE statistics=0.062, p=0.80441 >> looks Equality (fail to reject H0)
Satisfied with homogeneity of variance.

08: 게시판에 있는 레시피에 대한 다른 사용자의 평가가 레시피 선택에 도움이 되었나요?
LEVENE statistics=7.000, p=0.01322 >> does not look Equality (reject H0)
Unsatisfied with homogeneity of variance, Go to Welch T Test

```

```
def callLevene(helper,thousand):
    stat, p = levene(helper,thousand)
    print(f'LEVENE statistics={stat:.3f}, p={p:.5f}',end = '')

    alpha = 0.05
    if p > alpha:
        print(f" >> looks Equality (fail to reject H0)")
        print("Satisfied with homogeneity of variance.\n")
        return True
    else:
        print(f" >> does not look Equality (reject H0)")
        print("Unsatisfied with homogeneity of variance, Go to Welch T Test\n")
        return False
```

[그림 78] 등분산성 검정을 위한 코드와 결과

9.3.4. 검정 결과

정규성과 등분산성 만족 여부에 따라 각각 다른 검정 방식을 채택하였고 그 결과 상황에 따라 알맞게 Independent T 검정과 Welch T 검정, Wilcoxon Rank Sum 검정을 시행하였다. 아래 코드는 검정을 위한 코드와 모든 문항에서 이를 수행하기 위해 자동화한 코드이다.

```
def doIndTTest(num,helper,thousand):
    stat, p = ttest_ind(helper,thousand)
    print(f'TEST statistics={stat:.3f}, p={p:.5f}\n')

    alpha = 0.05
    if p > alpha:
        print(f"통계적으로 유의미하지 않음(fail to reject Null Hypothesis)\n")
        normal.append(num)
    else:
        print(f"통계적으로 유의미함(reject Null Hypothesis)\n")
        not_normal.append(num)
```

[그림 79] Independent T test, Welch T test 코드

```

def doRankSumTest(num,helper,thousand):
    stat, p = ranksums(helper,thousand)
    print(f'TEST statistics={stat:.3f}, p={p:.5f}')

    alpha = 0.05
    if p > alpha:
        print(f"통계적으로 유의미하지 않음(fail to reject Null Hypothesis)\n")
        normal.append(num)
    else:
        print(f"통계적으로 유의미함(reject Null Hypothesis)\n")
        not_normal.append(num)

```

[그림 80] Wilcoxon rank-sum test 코드

```

def Ttest(num,helper,thousand):
    print("Cal Shapiro...")
    if calShapiro(helper,thousand):
        print("Cal Levene...")
        if calLevene(helper,thousand):
            print("== Independent t-test ==")
        else:
            print("== Welch t-test ==")
        doIndTTest(num,helper,thousand)

    else:
        print("== Wilcoxon Rank-sum Test ==")
        doRankSumTest(num, helper, thousand)

```

[그림 81] 정규성 검사, 등분산성 검사, T 검정 자동화 코드

9.4. So what Questions에 대한 응답

결과를 도출하기 위해 T검정을 수행한 문항은 Q6, Q7, Q9, Q10, Q11, Q13, Q14이다. 이 중 Q13은 최종적으로 T검정에서 결과가 유의미한 차이가 나지 않는다고 판단되어 결과 분석에서 제외하였다. 다음은 각 문항에 대한 분석 결과이다.

Q6: 0) 어플리케이션이 본인이 생각한 목적 달성에 도움이 되었나요?

ANS_helper [5 3 5 5 5 5 4 4 3 4 3 5 5 5]

ANS_thousand [3 4 4 3 4 5 4 4 4 4 4 3 4 3]

Cal Shapiro...

HELPER statistics=3.148, p=0.20723 >> looks Gaussian (fail to reject H0)

THOUSAND statistics=0.360, p=0.83517 >> looks Gaussian (fail to reject H0)

Satisfied with normality.

Cal Levene...

LEVENE statistics=1.155, p=0.29175 >> looks Equality (fail to reject H0)

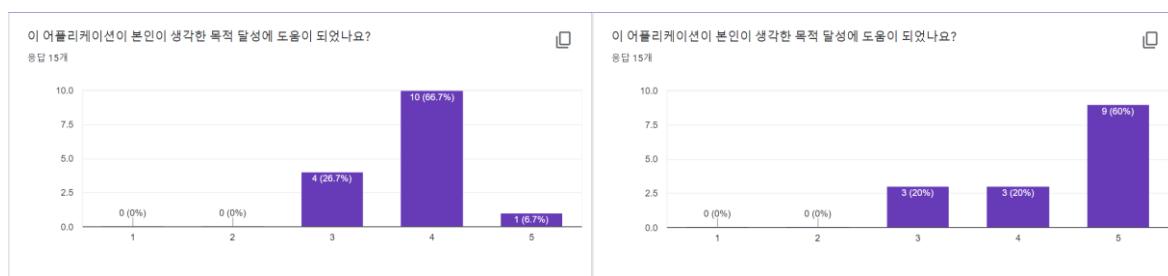
Satisfied with homogeneity of variance.

== Independent t-test ==

TEST statistics=2.324, p=0.02762

통계적으로 유의미함(reject Null Hypothesis)

[그림 82] Q6 분석 결과



[그림 83] Q6 설문 결과 (좌) 만개의 레시피 (우) Recipe Helper

Q6의 분석결과 목적 달성에 있어서 유의미한 차이가 있었고, '만개의 레시피'와 'Recipe Helper' 모두 사용자가 생각한 목적을 달성할 수 있는 쓸모 있는 어플이었지만, 'Recipe Helper'의 만족도가 더 높은 것으로 판단하였다.

Q7: 최종적으로 레시피를 선택하는 데 얼마나 많은 시간이 걸리나요?

ANS_helper [3 2 3 3 2 1 3 3 2 3 2 3 3 1 1]
ANS_thousand [4 3 5 4 3 2 3 4 4 4 4 4 5 3 5]

Cal Shapiro...

HELPER statistics=2.823, p=0.24380 >> looks Gaussian (fail to reject H0)
THOUSAND statistics=0.392, p=0.82199 >> looks Gaussian (fail to reject H0)
Satisfied with normality.

Cal Levene...

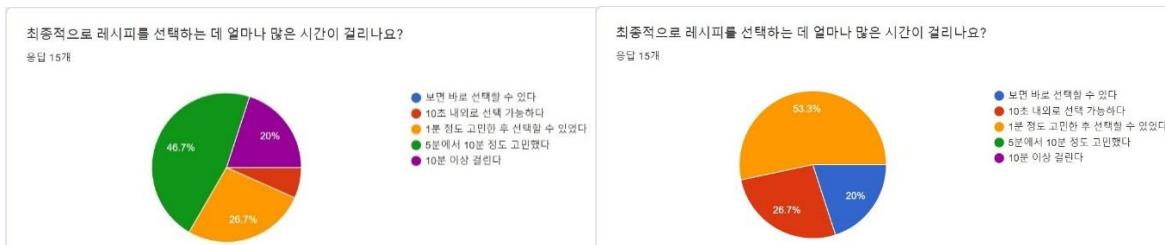
LEVENE statistics=0.062, p=0.80441 >> looks Equality (fail to reject H0)
Satisfied with homogeneity of variance.

== Independent t-test ==

TEST statistics=-4.785, p=0.00005

통계적으로 유의미함(reject Null Hypothesis)

[그림 84] Q7 분석 결과



[그림 85] Q7 설문 결과 (좌) 만개의 레시피 (우) Recipe Helper

Q7의 분석 결과 레시피 선택 시간에 있어서 유의미한 차이가 있었다. '만개의 레시피'의 경우 레시피를 선택하는데 5분 이상 걸리는 경우가 66% 이상이었고, 'Recipe Helper'는 1분 이내로 레시피를 고를 수 있음이 100%로 모든 사용자가 빠른 시간 내에 레시피를 선택하는데 도움을 주었음을 알 수 있다.

Q9: 다양한 재료를 활용할 수 있도록 레시피를 제공했나요?

ANS_helper [5 3 5 4 5 5 5 5 4 4 4 3 5 5 5]
ANS_thousand [2 3 3 3 4 4 4 5 2 2 2 2 1 3 1]

Cal Shapiro...

HELPER statistics=3.407, p=0.18206 >> looks Gaussian (fail to reject H0)
THOUSAND statistics=0.406, p=0.81620 >> looks Gaussian (fail to reject H0)
Satisfied with normality.

Cal Levene...

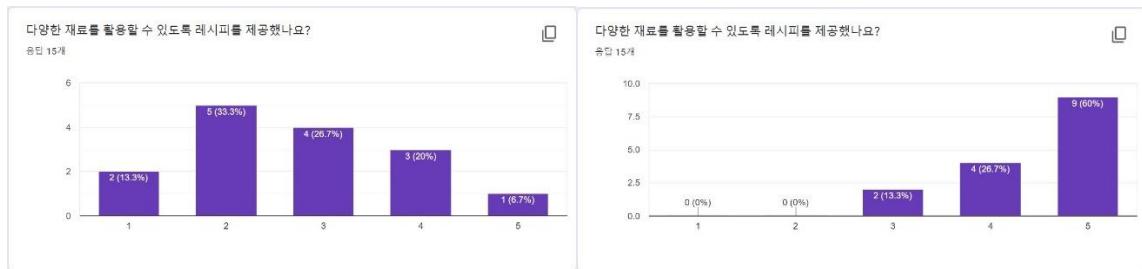
LEVENE statistics=2.291, p=0.14134 >> looks Equality (fail to reject H0)
Satisfied with homogeneity of variance.

== Independent t-test ==

TEST statistics=4.864, p=0.00004

통계적으로 유의미함(reject Null Hypothesis)

[그림 86] Q9 분석 결과



[그림 87] Q9 설문 결과 (좌) 만개의 레시피 (우) Recipe Helper

Q9의 분석 결과 다양한 재료 활용에 있어서 유의미한 차이가 있었다. '만개의 레시피'의 경우 재료와 관련하여 레시피를 제공하는 일이 극히 드물었지만, 'Recipe Helper'는 대체로 식재료를 활용할 수 있게 레시피를 제공함을 알 수 있었다. 이런 결과나 나온 이유는 'Recipe Helper'에는 식재료로 레시피를 검색할 수 있는 방법이 있기 때문으로 판단하였다.

Q10: 조리법, 영양 성분, 필요한 재료 등의 레시피에 대한 정보가 상세하게 제공되었나요?
ANS_helper [5 3 5 3 5 5 4 4 4 4 3 4 5 5]
ANS_thousand [2 3 5 3 5 4 4 4 2 2 2 2 2 1]

Cal Shapiro...

HELPER statistics=2.528, p=0.28253 >> looks Gaussian (fail to reject H0)
THOUSAND statistics=1.743, p=0.41842 >> looks Gaussian (fail to reject H0)
Satisfied with normality.

Cal Levene...

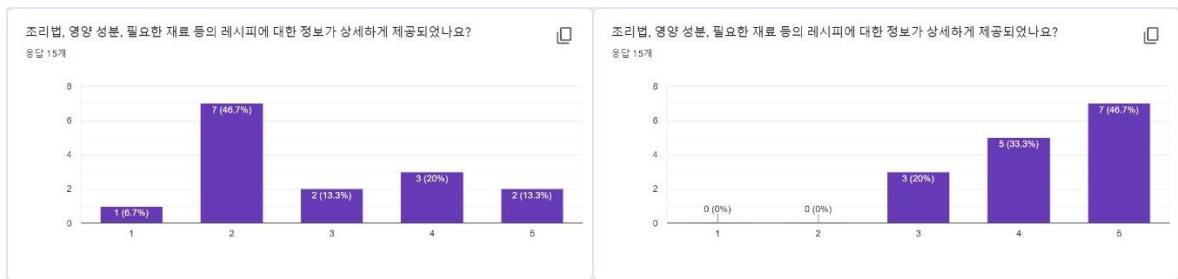
LEVENE statistics=1.094, p=0.30459 >> looks Equality (fail to reject H0)
Satisfied with homogeneity of variance.

== Independent t-test ==

TEST statistics=3.664, p=0.00103

통계적으로 유의미함(reject Null Hypothesis)

[그림 88] Q10 분석 결과



[그림 89] Q10 설문 결과 (좌) 만개의 레시피 (우) Recipe Helper

Q10의 분석 결과 레시피 외의 정보 제공에 있어서 유의미한 차이가 있었다. ‘만개의 레시피’와 ‘Recipe Helper’ 모두 조리법과 필요한 재료는 제공해주지만 영양 정보는 ‘Recipe Helper’에서만 제공한다. 이와 같은 결과를 통해 ‘Recipe Helper’는 사용자에게 대부분의 레시피에서 영양 정보를 제대로 제공하는 것을 판단할 수 있었다.

Q11: 이 어플리케이션을 활용해 고른 레시피가 실제로 요리할 때 도움이 됩니까?

ANS_helper [5 4 5 4 5 5 5 4 5 3 4 3 4 5 5]

ANS_thousand [3 3 4 3 4 4 3 3 4 3 4 4 4 4 4]

Cal Shapiro...

HELPER statistics=2.249, p=0.32477 >> looks Gaussian (fail to reject H0)

THOUSAND statistics=15.516, p=0.00043 >> does not look Gaussian (reject H0)

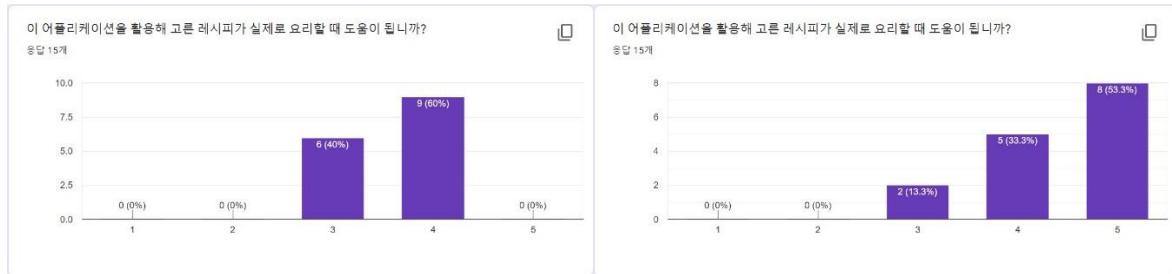
Unsatisfied with normality, Go to Wilcoxon Rank Sum Test

== Wilcoxon Rank-sum Test ==

TEST statistics=2.738, p=0.00619

통계적으로 유의미함(reject Null Hypothesis)

[그림 90] Q11 분석 결과



[그림 91] Q11 설문 결과 (좌) 만개의 레시피 (우) Recipe Helper

Q11의 분석 결과 요리를 할 때 주는 도움에 있어서 유의미한 차이가 있었다. '만개의 레시피'와 'Recipe Helper' 모두 요리에 큰 도움이 되지만 'Recipe Helper'가 5점에 많은 이유는 '요리를 선택할 때 활용성이 더 좋았기 때문에'로 판단하였다.

Q13: 어플리케이션의 기능을 사용하는 과정이 복잡했나요?

ANS_helper [1 2 1 1 5 3 2 3 1 2 2 3 1 1 5]

ANS_thousand [4 2 1 3 1 2 2 5 3 4 3 3 3 3 4]

Cal Shapiro...

HELPER statistics=4.016, p=0.13427 >> looks Gaussian (fail to reject H0)

THOUSAND statistics=0.009, p=0.99574 >> looks Gaussian (fail to reject H0)

Satisfied with normality.

Cal Levene...

LEVENE statistics=0.412, p=0.52629 >> looks Equality (fail to reject H0)

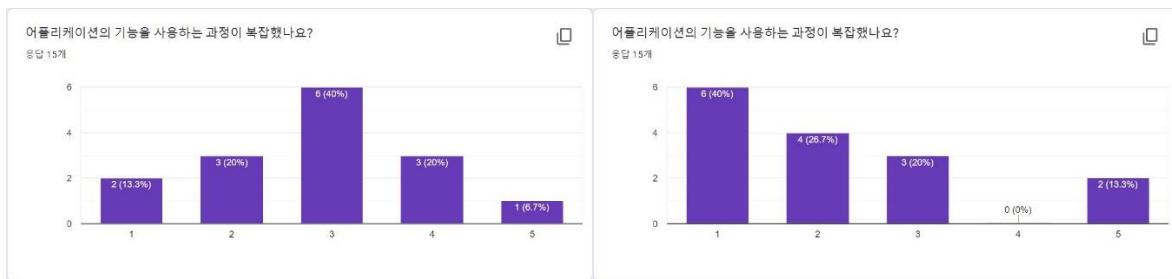
Satisfied with homogeneity of variance.

== Independent t-test ==

TEST statistics=-1.454, p=0.15700

통계적으로 유의미하지 않음(fail to reject Null Hypothesis)

[그림 92] Q13 결과 분석



[그림 93] Q13 설문 결과 (좌) 만개의 레시피 (우) Recipe Helper

Q13의 분석 결과 기능의 복잡도에 있어서 유의미한 차이가 없었다. p-value가 0.05보다 크기 때문에 귀무 가설을 기각하지 못했다. 유의미한 차이가 없기 때문에 더 이상 분석을 진행하지 않았다.

Q14: 필요한 기능을 직관적으로, 쉽게 찾을 수 있었나요?

ANS_helper [5 4 5 5 5 5 4 5 4 5 3 5 5 5]

ANS_thousand [2 4 5 3 3 4 3 5 2 3 3 3 3 2]

Cal Shapiro...

HELPER statistics=11.578, p=0.00306 >> does not look Gaussian (reject H0)

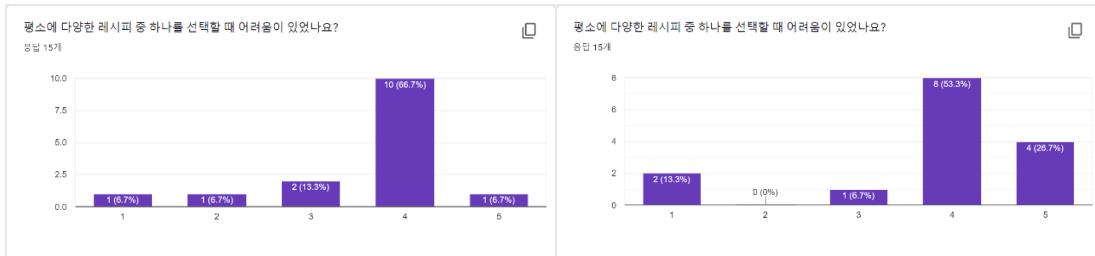
Unsatisfied with normality, Go to Wilcoxon Rank Sum Test

== Wilcoxon Rank-sum Test ==

TEST statistics=3.505, p=0.00046

통계적으로 유의미함(reject Null Hypothesis)

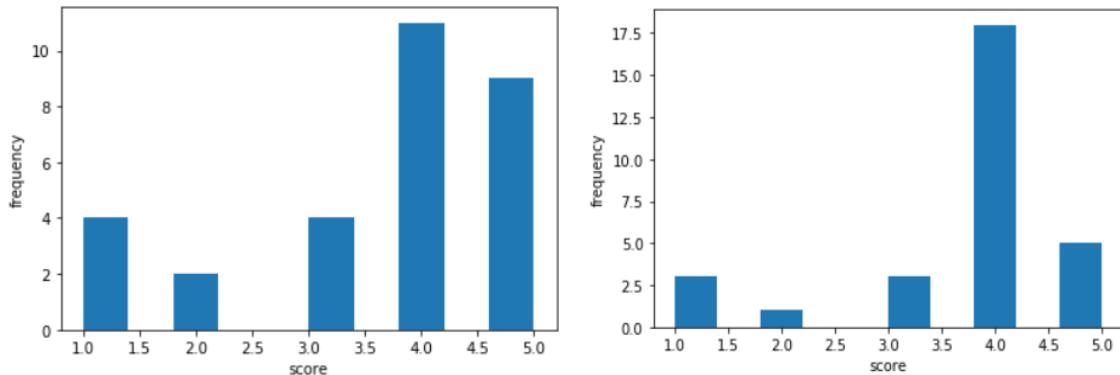
[그림 94] Q14 결과 분석



[그림 95] Q14 설문 결과 (좌) 만개의 레시피 (우) Recipe Helper

Q14의 분석 결과 UI/UX의 직관성에 있어서 유의미한 차이가 있었다. '만개의 레시피'와 'Recipe Helper' 모두 직관적으로 어떤 기능인지 알 수 있었지만 좀 더 단순하게 기능이 구성된 'Recipe Helper'의 UI가 사용성 면에서 더욱 직관적이라고 판단했다.

9.4.1. 결론



[그림 96] 문항 3 설문결과(좌), 문항 4 설문결과(우)

문항3의 질문은 "식재료가 남아서 번거로웠던 경험이 있는가?"이고, 문항4의 질문은 "레시피를 선택할 때 어려움을 겪은 경험이 있는가?"였다. 두 그룹의 설문 결과를 모두 합쳐 보았을 때 약 70~80%의 사용자들이 그렇다고 응답하였고, 이 결과는 "Recipe Helper"와 같은 추천 시스템이 많은 사용자들에게 도움이 될 수 있는 어플리케이션이라는 점이 명확해졌다.

결과적으로, 'Recipe Helper'가 기존 시스템의 대표 격인 만개의 레시피보다 대부분의 기능에서 더 높은 만족도를 보여주고 있고, 특히 기존에는 없던 서비스나 문제점을 개선한 부분에서 사용자들의 만족도는 유의미한 차이가 더욱 크게 두드러졌다. 이를 통해 TomTom VIP팀은 현재 상황에 대한 문제점을 제대로 직시하고 분석하여 잘 해결하였다는 것을 통계적인 수치로 알 수 있었다.

다만, 사용자 스터디를 진행하다 보니 기존에 기획하였던 사용자 스터디 참여자 50명을 모두 모으지 못하고 각 서비스 당 15명씩 총 30명을 참여자를 모집할 수 있었던 점이 아쉬운 점으로 남았다. 이러한 이유는 코로나-19라는 상황과 반 학기밖에 되지 않는 시공간적인 제약과 더불어 바코드 마케팅을 제대로 펼치지 못해서라고 생각한다. 이 외에도 3가지의 아쉬운 점이 있었다.

첫째, 적은 User수와 지인 중심 그룹으로 이루어진 스터디였다는 점이다. 일중 수 이상의 사용자가 확보되어야 일반 대중 사이에서의 만족도를 알 수 있는데, 30명 정도의 지인 중심의 그룹으로 사용성 평가가 이루어지다 보니 대부분 비슷한 학력수준과 생활수준을 보였다. 따라서 일반적인 사용자 그룹이라고 보기에는 어폐가 있었으며, 지인들 사이에서 사용성 평가를 수행하였기 때문에 타당한 이유 없이 긍정적인 평가를 받았을 가능성이 높다. 이를 해결하기 위해서는 향후 더 광범위하고 적극적인 사용자 스터디를 진행해야 할 것으로 보인다.

둘째, 사용자 스터디가 짧은 기간동안 이루어진 점이다. 일주일 남짓의 기간 동안 사용자들이 어플리케이션에서 발생할 수 있는 모든 상황을 평가하기는 어렵다. 또한 매일 요리를 하지 않는 사용자도 있다 보니 충분한 사용 후 설문을 진행하지 않았을 가능성이 있다. 만약 기간을 늘려서 다시 사용성 평가를 수행한다면 버그를 발견할 수도 있고, 불편한 점을 발견할 수도 있다. 이렇게 문제점이 발생한다면 평가가 달라질 수도 있을 것으로 보인다.

셋째, 사용자가 실제로 어플리케이션을 사용하였는지 알 수 없는 점이다. 설문 조사는 어플 내부에서 실행할 수 있지만 외부에서도 진행할 수 있다. 따라서 실제로 설문 참여자가 어플리케이션을 설치하고 적절한 기간을 사용한 이후에 설문을 진행하였는지 확실하게 알 수 없다. 따라서 사용자 평가 결과의 신뢰도에 의심을 가질 수 있다.

10. Issue and Solution

프로젝트를 진행하면서 수많은 문제점들에 부딪혔는데 이 문제점을 아래 항목에서 자세히 살펴볼 수 있다. 문제점을 어떻게 해결하였는지 또한 자세히 설명한다.

10.1. 추천 알고리즘 데이터셋 구축

TomTom VIP팀이 설계한 추천 알고리즘은 사용자의 사용이력, 개인정보 등을 활용한다. Collaborate Filtering과 Content Based Filtering을 결합한 Hybrid Filtering 추천 모델은 많은 양의 사용자의 history와 score가 필수적이지만, 프로젝트의 기간이 매우 짧았다. 사용자가 많을수록, 사용기간이 길수록 데이터가 축적되어 더 좋은 추천 모델을 만들 수 있지만 초기 데이터도 없어서 추천 모델이 제대로 작동하지 않았다. 이런 Cold Start 문제를 해결하기 위해 회원가입 시 사용자 취향 분석을 시작하고, 이 자료를 history로 사용하였다. 또한 적은 history에도 만족할 만한 추천 시스템을 만들기 위해 scrap 항목과 history의 가중치를 다르게 두어 사용자가 더 관심을 가질 것 같은 레시피들을 추천하였다. 단기간 프로젝트였기에 사용자 확보와 데이터 축적에 어려움이 있었지만 위와 같은 방법으로 잘 해결할 수 있었다. 만약 어플리케이션을 지속적으로 발전시켜 나간다면 사용량이 증가하여 추천 알고리즘 성능 역시 개선될 것으로 예상한다.

10.2. 무료 서버

본 프로젝트를 진행하면서 서버는 Naver Cloud에서 제공하는 무료 서버를 사용하였다. 때문에 $100,000 * 100,000$ 연산은 6200시간으로 현실적으로 불가능하였고, 분류한 음식들 중 가장 개수가 적은 $500 * 500$ 으로 연산을 시도했지만 이 역시 15~20초로 상당히 느렸다. Trade off 관계인 성능과 수행시간을 적절히 조율하여 최종적으로 $100 * 100$ 연산을 통해 레시피 추천에 최대 3초가 소용되도록 시스템을 수정하였다. 레시피별로 카테고리를 만들어 사용자의 관심도가 높은 레시피의 카테고리를 선택하고 해당 카테고리 내에서 100개를 임으로 선정하여 추천 알고리즘에 입력하는 방식으로 시스템이 재설계되었다.

10.3. 한국어의 불규칙성과 불완전성

'만개의 레시피'에서 크롤링한 레시피의 식재료 데이터를 분석하면서 재료의 표기에 문제가 있음을 발견하였다. '만개의 레시피'에 있는 레시피 역시 일반 사용자가 업로드하는 방식이기 때문에 하나의 단어를 나타내는 다양한 단어가 존재하기도 했고, 비표준어를 사용하기도 하여 모델 학습이 제대로 되지 않았다. 따라서 이를 정제하는 과정이 필요했다.

하나의 단어를 나타내는 다양한 방식의 예로는 맛술을 미림, 미향, 미작, 미정, 조미용 마술 등

다양한 형태로 나타내는 것이다. 비표준어를 사용하는 예는 허니 머스터드를 허니 머스타드, 허니 머스타드, 허니머스터드 등의 형태로 나타내는 것이다. 이 같은 이유 때문에 식재료의 개수는 100,000개의 레시피에서 72,000개 이상이 나왔다. 따라서 자주 나오는 식재료 상위 3000개를 기준으로 표준화 작업을 진행하여 상위 3000개의 식재료를 362개로 줄일 수 있었다.

10.4. 레시피 정보의 부정확함

제육볶음a		
재료	Ingredients	
[재료]		
돼지고기 제육볶음용 ①	600g	

제육볶음b		
[제육볶음양념장]		
재료	Ingredients	
제육볶음재료		
돼지고기앞다리살 ①	1근	물엿 ①
양파 ①	1개	매실액 ①
당근 ①	1/3개	고추장 ①
마늘 ①	1스푼	고춧가루 ①
파 ①	약간	후추가루 ①
고추 ①	기호에맞게	약간

[그림 97] 유사한 레시피의 식재료

마찬가지로 '만개의 레시피'는 일반 사용자가 레시피를 업로드 하기 때문에 정해진 규약이 없다. 따라서 같은 요리임에도 불구하고 재료 정보가 같지 않은 경우도 있는데 이런 정보 때문에 레시피별 유사 관계가 제대로 계산되지 않았다. 또 다른 문제로 유사한 요리임에도 간장, 고추장 등 부재료 및 조미료를 식재료로 넣는 경우도 있고 그렇지 않은 경우도 있어서 역시 유사 관계가 제대로 계산되지 않았다. 이를 해결하기 위해 기존의 Euclidean-d 문제를 해결하였다.

10.5. 기준 Euclidean-d의 문제

10.4.에서 언급한 첫 번째 문제의 예는 아래 표와 같다.

레시피 명	간장	고추장	돼지고기	더덕	양파	소금
제육볶음1	1	0	1	0	0	0
제육볶음2	1	1	1	0	1	1
더덕구이	1	1	0	1	1	1

[표 4] 첫 번째 문제의 예시

[표 4]에서 <제육볶음1-제육볶음2>의 distance는 $D=\sqrt{3}0$ 이 <제육볶음1-더덕구이>의 distance는 $D=\sqrt{2}$ 로 제육볶음1이 제육볶음2보다 더덕구이와 더 유사하다는 결론이 나왔다. 이 때, Euclidean distance가 더 짧을수록 유사하다고 측정한다.

10.4.에서 언급한 두 번째 문제의 예는 아래 표와 같다.

레시피 명	간장	고추장	돼지고기	더덕	양파	소금
불고기 구이	0	0	1	0	0	0
제육볶음	1	1	1	0	1	1
더덕구이	1	1	0	1	1	1

[표 5] 두 번째 문제의 예시

[표 5]에서 <제육볶음-불고기>는 $D=\sqrt{4}$ 가 <제육볶음-더덕구이>는 $D=\sqrt{2}$ 가 나왔다. 이를 통해 제육볶음이 불고기보다 더덕구이 가 더 유사하다는 결론이 나온다.

따라서, 300여개의 재료 중 레시피의 정체성을 결정하는 주재료를 선정하여 가중치를 조정하였고 이를 통해 결과적으로 레시피 간 유사도를 더욱 정확하게 측정할 수 있었다. 가중치 조정을 통해 <제육볶음-불고기>의 distance를 계산하면 $D=\sqrt{4}$ 가 나오고 <제육볶음-더덕구이>의 distance는 $D=\sqrt{50}$ 이 나와 더욱 정확하게 유사도가 산정되었다.

레시피 명	간장	고추장	돼지고기	더덕	양파	소금
불고기 구이	0	0	5	0	0	0

제육볶음	1	1	5	0	1	1
더덕구이	1	1	0	5	1	1

[표 6] 가중치를 조정한 결과 예시

10.6. 레시피에 대한 영양 정보

'만개의 레시피'에서 식재료별로 정확한 수량을 적지 않는 경우가 많기 때문에 기존에 식재료를 이용한 칼로리 계산은 어려워졌다. 따라서 식재료 대신 음식명을 이용해 칼로리를 산출하기로 하였지만 식약청에서 제공하는 음식 종류는 턱없이 적었다.

결과적으로 레시피에 대한 영양정보는 수천만명의 사용자를 보유한 식단 관리 어플인 Fatsecret이 제공하는 음식 영양 정보 open api를 통하여 계산하였다. 해외 api여서 한국어 검색을 지원하지 않아 Papago api도 함께 사용하여 음식명을 영어로 번역하고 그 값을 다시 fatsecret에 전달하는 방식으로 영양 정보를 구하였다.

11. Reflection

본 프로젝트를 진행하면서 아쉬웠던 점과 느낀 점은 아래와 같다.

11.1. 아쉬웠던 점

모든 팀이 그렇겠지만 코로나-19로 인해 비대면으로 진행하였던 점이 가장 아쉬웠다. 만나서 의견을 나누는 것이 더 원활하기도 하고 모르는 것이 있으면 알려주는 면에서도 대면으로 프로젝트를 진행하는 것이 훨씬 효율적이었을 것 같아서, 만날 수 있는 상황이 자주 있다면 더 완성도 높은 프로젝트가 되었을 것 같다. 그리고 반 학기라는 짧은 시간과 적은 인원으로 아이디어 구상부터 기획, 개발을 진행했기에 100% 완벽한 어플리케이션을 만들지 못한 점 또한 아쉬웠다. 더하여 추천 시스템을 처음 접했는데 이와 관련된 자료가 풍부하지 않아 더 좋은 추천 시스템을 만들지 못한 점이 계속 신경 쓰여 기회가 된다면 추천 분야는 다시 도전해보고 싶었다.

11.2. 느낀 점

프로젝트를 진행하며 HCI수업때 배운 내용을 적용해볼 수 있어서 좋았다. 단순히 배우는 것 보다는 개발에 직접 적용하고 경험해보는 것이 '왜 해야하는지'와 '왜 중요한지'에 대해 더 잘 이해할 수 있게 했다. 또한 안드로이드, SQL, 머신러닝 모델 구축, API 활용 등 시스템을 만들기 위해 필요한 전반적인 분야의 지식을 쌓고 도전해보는 좋은 기회였다. 기존의 전공으로는 경험해보지 못하는 것들을 통해 팀원과 문제를 해결해 가는 과정의 즐거움과 협업 능력을 키울 수 있던 좋은 시간이었다.

다양한 배움 중 특히 기억에 남는 것은 추천 모델 구축과 사용자 스터디였다. 좋은 성능의 모델을 만들기 위해서는 훌륭한 전처리 과정이 필요한데 이를 수행하는 것이 쉽지 않았다. 많은 노력도 필요하고 머리도 많이 써야 하는 부분이라 모델을 만들면서 많은 애를 먹었었다. 다양하게 표현되어 있는 단어들을 하나의 단어로 바꾸는 것은 결코 쉬운 일이 아니었고 그 양 또한 매우 많아서 더욱 힘들었던 것 같다. 팀원들이 모두 함께 힘써줘서 이렇게 좋은 결과를 얻어낼 수 있었던 것 같다. 또한 주 재료에 가중치를 주는 방법도 매우 신선한 발상이었던 것 같다. 덕분에 유클리드 거리를 구할 때 더욱 좋은 성능을 끌어낼 수 있었고, 사용자에게 더 알맞은 추천을 해줄 수 있었다. 사용자 스터디는 처음해보는 일이라 어떻게 질문을 만드는 것이 좋을지 잘 몰랐었는데, 비교군을 만들어 독립적으로 질문해야 한다는 것과 그 질문을 통해 "So what?"이라는 질문에 자답할 수 있도록 문항을 설계해야 좋은 사용자 스터디를 진행할 수 있다는 것을 알게 되어 큰 도움이 되었다. 또한 올바른 결론을 내기 위해서는 검정 절차를 거쳐야 한다는 것도 좋은 배움이 되었다. 한 팀원은 이번학기에 "다학제 캡스톤 디자인"을 같이 수강하였는데 HCI수업 때 배운 사용자 스터디를 활용해 원활하게 사용자 스터디를 진행할 수 있었다.

프로젝트를 성공적으로 마치기 위해 같이 헌신하고 힘써준 동료들에게 큰 박수를 보낸다.