**서비스 데이터 사이언스  
FINAL REPORT**

**- 반려식물 추천 서비스 -**

<https://drive.google.com/file/d/1TpILlWTY4JvDGrzKAWitwwrp8GMrki8O/view?usp=sharing>

TEAM 5

산업경영공학과 2019100889 이수진

산업경영공학과 2019100907 조희수

산업경영공학과 2019100866 노희지

산업경영공학과 2019100869 문선웅

목차

**제 1 장 개발기술의 개요**

제 1 절 기술개발 기획의도 및 목적

1.1.1. 기획 의도

1.1.2. 기획 목적

제 2 절 기술개발 현황 및 배경

**제 2 장 개발목표 및 개발 내용**

제 1 절 기술개발 목표

2.1.1. 최종 목표

2.1.2. 기능 기술개발 목표

2.1.3. 기술개발 목표 변경사항

제 2 절 세부 개발내용 및 방법

2.2.1. 기술개발 추진 방법 및 전략

2.2.2. 기술개발 주요메뉴 구조

2.2.3. 기술개발 환경

2.2.4. 기술개발 내용

**제 3 장 성과요약 및 기대효과**

제 1절 핵심 개발성과 및 독창적 핵심기술

제 2절 기대효과

제 3절 마케팅 전략 및 향후 추진 계획

**제 4장 소스 코드 출처 및 참고문헌**

제 1절 소스 코드 출처 및 사용 범위

제 2절 참고 문헌

**제 1 장 개발기술의 개요**

제 1 절 기술개발 기획의도 및 목적

* + 1. 기획 의도

최근 반려동물에 이어 반려식물이라는 개념이 부상하며 식물에 대한 관심이 커지고 있다. [1]반려식물이란 사람이 정서적으로 의지하고자 가까이 두고 기르는 식물을 일컫는 말이다. 인간의 행복과 같은 심리정서적인 부분에 있어 동물뿐만 아니라 식물까지도 긍정적인 영향을 미친다는 점에서 반려식물이라는 개념이 만들어진 것이다.

시장조사전문기업 엠브레인 트렌드모니터에 따르면, 전국 만 19~59세 성인남녀 1000명을 대상으로 반려식물과 관련해 설문 조사를 실시한 결과 전체 응답자의 58%가 직접 식물을 키우고 있는 것으로 나타났다. 27%는 현재 키우지는 않지만 과거에 키워본 적이 있다고 응답했고, 한번도 식물을 키워본 적이 없다는 응답은 단 15%에 그쳤다. 대부분의 사람들이 현재 직접 식물을 키우고 있거나 키운 경험을 가지고 있었으며, 식물을 소중하게 대하는 태도가 점차 강해지고 있는 것으로 조사되었다.

특히나 코로나19가 장기화되면서 사람들 사이 우울감과 무기력함을 느끼는 코로나 블루 현상이 확산되고 있는데, 이를 위한 치료제로 사람들은 반려 식물을 찾고 있다. 또한 대인 활동이 줄어들고 집에서 보내는 시간이 늘어나면서 언택트 라이프에 맞는 새로운 재미 요소로서 반려 식물을 찾기도 한다. [2]실제 온라인 쇼핑몰 인터파크에 따르면 코로나 장기화가 시작됐던 2020년부터 식물 관련 가드닝 상품군의 매출은 지난해보다 30% 이상 증가했다. 이외에도 모종과 흙 매출이 각각 92%, 88% 증가했고 화분 매출은 48%가 증가했다. 비교적 적은 비용으로 정서적 위안과 심리적 안정감을 얻을 수 있다는 점에서 식물은 큰 장점을 가지고 있어 더욱 사람들의 관심이 쏠리고 있는 것을 확인해 볼 수 있다.

하지만 이러한 반려 식물도 마냥 키우기 쉬운 것만은 아니다. [3]실제로 식물 관리에 대한 인식 조사 결과 식물을 키우고 있거나 과거 키웠던 경험이 있는 사람들 중 절반 이상의 사람들이 식물을 키울 때 ‘관리가 어렵다’라고 답했다. 또한 과거에 식물을 키우지 않았지만 현재 키우지 않는 사람들 역시 이유는 ‘관리가 힘들어서’라는 답변이었다. 그래서인지 사람들은 키우기 쉬운 반려식물을 고르기 위해 블로그, 뉴스, 트위터 상에도 반려식물 추천 등의 키워드가 검색되고 있으며, 유튜브 상에도 초보자를 위한 반려식물과 같은 콘텐츠도 올라오고 있다.

이처럼 반려식물에 대한 사람들의 관심이 증가함에 따라 반려식물을 추천해 주길 바라는 사람들의 요구가 더욱 커지고 있음을 발견할 수 있었다. 하지만 아직까지 추천 콘텐츠 수준에 그치고 있으며 추천의 기준 또한 단순히 손이 덜 가고 기르기 쉬운 것에 초점이 맞춰져 있는 것으로 보인다.

우리는 사람과 더욱 밀접해지고 있는 반려 식물 트렌드에 주목하여, 단순히 키우기 쉬운 반려식물이 아닌 키우는 사람의 환경과 취향에 알맞는 반려식물 추천 서비스를 선보이고자 한다.

* + 1. 기획 목적

반려식물에 대한 사람들의 관심이 높아지고, 해당 시장이 더욱 커지고 있음에 따라 우리의 서비스를 통해 반려식물 입문자에게 느껴질 수 있는 진입 장벽을 낮춰줄 뿐만 아니라 반려식물을 선택하고자 하는 모든 사람들에게 환경과 취향에 맞는 더욱 맞춤화된 반려식물을 추천해주는 것이 목적이다.

제 2 절 기술개발 현황 및 배경

 현재 반려식물 관련 앱 기술 시장을 살펴본 결과 이미 반려 식물을 키우고 있는 사용자들을 위한 반려 식물 관리 위주의 기술에 초점이 맞춰져 있었다.

앱 스토어 기준 반려 식물 키워드를 중심으로 대표적인 서비스들을 몇 가지 분석해보면, 첫번째로 “Plantgram” 이라는 서비스의 경우 커뮤니티 기술을 기반으로 하여 다른 사람들이 키우는 식물은 무엇인지, 어떻게 키우고 있는지 등 다양한 반려식물 생활을 사람들과 공유할 수 있도록 한다. 또한 사용자로 하여금 반려식물 정보를 등록 받아 물주기, 마지막 물 준 날, 등록일 등을 관리할 수 있도록 해주며 사람들이 등록한 반려 식물 데이터를 통해 인기순으로 반려식물들을 추천해주고 있다. 두번째 “그린그린” 서비스 또한 관리 단계에서의 편리성을 위해 개발된 서비스로 물주기 알림 기능을 비롯해 일정 관리를 위한 식물 캘린더 기능, 반려식물의 모습을 기록할 수 있는 식물 리포트 기능 등을 제공하고 있다. 해당 서비스에서는 사용자의 취향을 입력 받아 식물을 추천해주고 있다. 마지막으로 “싱그러운집” 서비스는 앞서 언급한 Plantgram과 비슷하게 커뮤니티 기술을 통해 반려식물 정보를 다른 사람들과 나누고, 태깅 기능을 이용해 원하는 식물과 상품에 쉽게 접근하여 구매까지 할 수 있도록 하는 서비스를 제공하고 있다. 이 서비스에서 사용자들에게 식물을 추천하는 방식은 1인 가구용, 선물용, 입문자용 등 카테고리에 따른 식물들을 추천해준다.

이처럼 현재 시장에 나와있는 반려식물 관련 서비스들의 추천 기술들은 단순히 다른 사람들이 많이 키우고 있는 인기도 기준, 혹은 서비스 자체에서 제공하는 카테고리에 따른 기준들을 가지고 추천해주고 있으며 사용자의 취향을 반영하기 위해서는 직접 데이터 입력이 필요한 상황이었다.

하지만 반려식물을 키우기 위해서는 식물 자체의 특성, 식물을 키우고자 하는 환경과 여건, 사용자의 취향과 관리 수준 등 다양한 요인들에 대한 이해와 고려가 필요하다. 따라서 단순 쿼리형 추천 시스템이 아닌 Machine Learning을 기반으로 개인 맞춤형 추천이 필요하며, 번거로운 인풋 과정 없이 간단한 좋아요 기능을 통해 좋아요를 누른 식물을 기반으로 소비자의 선호, 관심 식물을 파악해 그들이 좋아할 만한 제품 및 서비스를 자동으로 제공하는 편리성을 갖출 때이다.

**제 2 장 개발목표 및 개발 내용**

제 1 절 기술개발 목표

* + 1. 최종 목표

최종적인 목표는 사용자가 선택한 식물들을 기반으로 사용자에게 맞춤형 식물을 추천해주는 것이 궁극적인 목표이다. 따라서 맞춤형 식물을 추천해주기 위한 모델링과 모델링을 접목시켜 사용자에게 제공해줄 앱을 구현하는 것이 본 프로젝트의 목표라고 할 수 있다.

* + 1. 기능 기술개발 목표

먼저 모델링에서는 잎을 관상하는 식물과 건조에 강한 식물 두가지로 DB로 나눠서 모델링을 진행할 것이다. 각 DB별로 사용자에게 입력값으로 사용자가 선택하는 식물 번호를 받고 해당 번호들과 유사한 식물을 콘텐츠 기반 추천 알고리즘을 통해서 코사인 유사도 값을 기반으로 선택한 식물들과 유사한 식물들을 유사도 값이 높은 값부터 상위 n개까지 보여주고자 한다.

다음으로 앱 구현쪽 기술개발 목표는 식물 추천 정보를 제공하는 메인 화면에서 좌우 스크롤 기능을 구현하여 정보 탐색에 있어 편의성을 더욱 높이고자 한다. 이를 통해 사용자는 메인 화면에서 간단한 손동작만으로 추천 식물 정보들을 탐색할 수 있을 것이다. 상세 화면에서는 식물 특성 및 관리와 관련한 구체적인 정보들을 제공해주고자 하며, 좋아요 버튼 기능을 구현하여 사용자가 선호하는 식물 데이터를 얻고자 한다. 이렇게 얻은 식물 데이터를 기반으로 구현한 모델을 활용해 사용자에게 맞춤형 반려식물 정보를 메인 화면에서 추천해줄 것이다. 또한 사용자가 좋아요 누른 정보들은 Like 화면에서 모아볼 수 있도록 구현하는 것까지 기능 기술개발 목표로 삼았다.

* + 1. 기술개발 목표 변경사항

먼저 모델링 부문의 변경사항이다. 기존의 proposal에서는 모델링 작업 시 콘텐츠 기반 추천 알고리즘과 더불어 멀티암드 밴딧 알고리즘 또한 추가적인 모델링 작업으로 구현을 시도하고자 하였다. proposal에서 제시한 밴딧 알고리즘은 도박에서 사용되는 슬롯머신을 지칭하는 속어에서 이름을 가져온 알고리즘으로 둘 이상의 손잡이가 달려 있고 각 손잡이는 다른 속도로 돈을 지불하는 슬롯머신에서 따온 개념이다.  
 밴딧 알고리즘은 하이브리드 방식을 취하게 된다. A, C, B 순으로 확률이 높다면, 먼저 A가 확률이 가장 높게 나왔으므로 A를 더 자주 잡아당기는 선택을 하지만 B와 C를 당길 기회를 포기하지 않는다. A에서 계속해서 성과를 거두게 되면 그때 B와 C를 당길 기회가 A에게 더 주어져서 A를 더 자주잡아당기고, 만약에 C가 더좋아지고 A가 더 나빠지기 시작하면 처음에 A로가던 많은 기회를 C에게 돌린다. 그중 하나가 A보다 우수하고 이것이 초기 50번의 실험에서 감춰져 있었던 결과였다면, 더 많은 테스트를 통해 이 사실이 밝혀질수 있는 기회가 생기게 된 것이다.  
 이러한 개념을 기반으로 멀티암드 밴딧 알고리즘은 사용자가 열람하는 이벤트를 실시간으로 받아 열람 확률의 최대화를 계속해서 진행해보려 했으나 앱구현 및 모델링 작업시 사용자가 열람하는 이벤트를 실시간으로 받는 작업이 한계점으로 본 연구 분석가들이 판단했고 따라서 모델링 작업시 밴딧 알고리즘에 대해서 모델링 작업을 진행하지 않기로 계획을 변경했다.

다음은 앱 서비스 개발 부문의 변경사항이다. 전체적인 완성도를 높이기 위해 기존 목표로 삼았던 기능들 외에 검색 기능과 마이 기능을 추가하고자 한다. 검색 기능을 통해 사용자는 원하는 특정 식물에 대한 정보를 손쉽게 찾아볼 수 있도록 한다. 또한 개인 맞춤화에 초점을 맞춘 기술인 만큼 사용자 정보와 사용자가 키우는 반려 식물 정보를 관리할 수 있는 마이 기능까지 구현을 목표로 하고 있다.

제 2 절 세부 개발내용 및 방법

* + 1. 기술개발 추진 방법 및 전략

가. 콘텐츠 기반 필터링

**데이터 수집**

농사로 : 농촌진흥청 농업기술포털에서 잎을 관상하는 식물과 건조에 강한 식물 두가지 종류의 정보를 크롤링해 사용한다. 수집 시 식물을 키우는 데 필요한 정보 및 식물의 사진을 함께 수집하고자 하였다.

**카테고리화 진행을 통한 정보 벡터화**

기존 가지고 있는 데이터를 기준으로 칼럼 별 카테고리화를 진행하였다. 해당 카테고리를 숫자로 나타내어 벡터화를 진행하였다. 기존 데이터들은 텍스트로 이루어져 있기 때문에 같은 텍스트(카테고리)를 가진 데이터를 특정 숫자로 한정하여 벡터화를 진행하고자 하였다.

**코사인 유사도를 사용한 유사 특징 식물 추천**

식물이 가지고 있는 특징을 N차원에서 코사인 유사도를 사용해 거리가 가까운 식물을 추출하여 추천한다. 추천 시 벡터화된 데이터가 아닌 실제 데이터를 보여주며, 분석에 사용되지 않은 데이터도 함께 반환하고자 한다. 이 과정에서 유사도를 구하는데 필요한 칼럼만 존재하는 벡터화된 데이터셋과 사용자에게 직접 보여줄 식물 정보 데이터 셋이 요구된다.

나. Flutter 애플리케이션

**RootPage를 통한 페이지 제어**

BottomNavigationBar의 BottomNavigationItem을 선택했을 경우, 해당 Icon에 대응되는 페이지로 이동하게 하기 위해서 HomePage, LikePage, MyPage를 제어하는 RootPage를 만들어서 관리하는 전략이다.

**반복되는 위젯 활용**

HomePage의 추천하는 식물 정보를 제공하는 식물 카드인 RecomendPlantCard, 각각의 식물에 해당하는 세부 정보 페이지는 식물 이름, 식물 종류, 설명만 바뀌고 디자인이나 틀은 그대로 유지된다. 따라서 해당 페이지를 전부 수작업으로 구현하지 않고, final 변수 선언을 통해 parameter로 전달하고, required로 받아 페이지를 그리는 방식을 사용했다.

**화면을 분할한 뒤 각각의 파일로 저장**

페이지의 화면을 한 dart 파일에 구현하게 될 경우 가독성이 떨어지는 코드로 보일 수 있다. 따라서 각각의 Widget은 각각의 새로운 dart파일로 만들어 Container를 return하거나 Widget을 return하는 방식을 통해 실제 Page 구성은 가독성이 좋은 코드로 구성하는 전략이다.

* + 1. 기술개발 주요메뉴 구조

가. 콘텐츠 기반 필터링

**사용자가 좋아한 식물의 고유번호를 추출**

사용자가 좋아한 식물의 번호를 추출하는 함수이다. 사용자가 식물에 좋아요를 누르면 해당 식물의 번호가 반환된다.

**사용자 선호 식물의 특징 벡터를 사용한 컨텐츠 기반 필터링**

사용자가 좋아한 식물의 번호를 가지고 해당 식물이 잎을 관상하는 식물인지, 건조에 강한 식물인지를 구분한다. 그 후 각각의 식물 데이터셋 (잎을 관상하는 식물, 건조에 강한 식물) 에서 추출된 코사인 유사도 벡터에 맞는 유사도 값을 도출한다. 식물을 한가지만 선택하는 것이 아니므로 잎을 관상하는 식물과 건조에 강한 식물 각각에서 구한 추천 식물들을 유사도 순으로 재정렬하여 상위 10가지의 식물을 추천한다.

**식물의 설명을 추출**

추출된 상위 10가지 식물의 식물 번호를 가지고 벡터화되지 않은 사용자에게 보여줄 특징 데이터 셋을 불러온다. 해당 데이터셋은 식물의 특징이나 관리 방법, 병충해 관리 방법에 대해 서술되어 있다.

나. Flutter 애플리케이션

**WireFrame**

다음은 앱의 전체적인 와이어프레임이다.



[그림 1]

- Home: 피드 형태를 통해 사용자에게 추천하는 식물 리스트를 보여준다. 사용자의 좋아요 기록이 누적됨에 따라 해당 정보들을 바탕으로 더욱 맞춤화된 추천 리스트가 제공된다.

- 상세 정보 화면: 메인 화면에서 마음에 드는 식물 정보를 선택하면 해당 식물에 대한 상세 정보가 제공된다.

- Like: 사용자가 좋아요 누른 식물 리스트를 모아볼 수 있다. 여기서도 식물 정보를 선택하면 상세 정보 화면으로 넘어간다.

- My: 사용자의 프로필을 비롯해 반려식물 정보 관리 및 설정 화면이다.

**IA**

다음은 앱의 IA(Information Architecture) - 정보 구조도이다.

테이블이(가) 표시된 사진

자동 생성된 설명

[그림 2]

* + 1. 기술개발 환경

가. 콘텐츠 기반 필터링

**Programming Language**

개발 언어로는 Python을 사용해 알고리즘 구현을 하였다.

**IDE**

모델을 구축하기 위해 사용한 IDE로는 주피터 노트북을 사용하였다.

**Library**

웹 페이지에 접근하여 크롤링을 진행하기 위해 requests, BeautifulSoup4, urllib.request 를 사용하였으며, 데이터 전처리를 진행하기 위해 pandas, time, datetime, numpy를 사용하였다. 또한 코사인 유사도를 구하기 위해 사이킷런의 cosine\_similarity 메서드를 사용하였다.

나. Flutter 애플리케이션

**Programming Language**

개발 언어로는 구글이 내놓은 프로그래밍 언어인 dart를 사용하였고, 개발 프레임워크로 dart를 assemble해 주는 flutter를 사용하였다.

**IDE**

개발 IDE로 Android Studio를 사용하였다. Android Studio에서 Flutter와 Dart SDK 를 설치해 플러터 애플리케이션을 구현하였다. 구현 화면을 확인하기 위한 에뮬레이터로는 Phone 화면의 Pixel 2 XL 을 사용하였다. 안드로이드 OS로는 Pie를 선택하여 개발을 진행하였다.

**Tools**

기타 도구로는 구글의 firebase를 사용하였다. 백엔드와 프론트엔드 사이의 구성 요소들을 유지관리하고 운영하는 도구이다. 클라이언트, SDK와 상호작용하며 middleware과도 원활하게 통신이 이루어지도록 한다.

* + 1. 기술개발 내용

가. 콘텐츠 기반 필터링

**데이터 수집**

데이터 수집을 위해 농사로:농촌진흥청 농업기술포털(https://www.nongsaro.go.kr)에서 두가지 종류의 식물 정보를 크롤링 하였다. 식물의 종류는 잎을 관상하는 식물과 건조에 강한 식물 두가지가 존재한다.

html 형식의 페이지 구조에서 태그 검색을 사용해 데이터를 수집하였으며, 수집한 데이터는 식물 각각의 사진(jpg)파일, 식물 이름, 분류, 생육형태, 생장높이, 생장너비, 실내정원, 잎형태, 생태형, 잎무늬, 광요구도, 관리수준, 관리요구도, 향기, 독성, 비료, 토양, 물주기, 병충해, 병충해관리, 특별관리, 잎색, 꽃피는 계절, 꽃색, 열매색, 열매맺는 계절, 뿌리형태, 생장형 등이 있다. 데이터 수집 예시는 다음과 같다.

테이블이(가) 표시된 사진

자동 생성된 설명

[그림 3]

**데이터 전처리**

데이터는 텍스트로 이루어져 있고, 비교적 같은 값을 갖는 데이터가 많았기 때문에 같은 데이터를 갖는 데이터들을 숫자로 표현한다. 카테고리의 기준은 다음과 같다.

| **plant** | **colunms** | **values** | **category** |
| --- | --- | --- | --- |
| 잎을 관상하는 식물 | 광요구도 | 낮은 광도 | 0 |
| 중간+낮은 광도 | 1 |
| 중간광도 | 2 |
| 중간+높은광도 | 3 |
| 높은광도 | 4 |
| 낮은+중간+높은 광도 | 5 |
| 관리수준 | 전문가 | 2 |
| 경험자 | 1 |
| 초보자 | 0 |
| 관리요구도 | 낮음 | 3 |
| 보통 | 2 |
| 필요 | 1 |
| 특별관리 | 0 |
| 생육적온 | 10~15 | 4 |
| 16~20 | 3 |
| 21~25 | 2 |
| 3~9 | 1 |
| -1이하 | 0 |
| 물주기 | 토양 표면이 말랐을 때 | 1 |
| 항상 촉촉하게 | 2 |
| 화분이 대부분 말랐을 때 | 0 |
| 항상 물에 잠기게 | 3 |
| 비료 | 적게 | 0 |
| 보통 요구 | 1 |
| 많이 요구 | 2 |
| 토양 | 중성 | 0 |
| 산성 | 1 |
| 중성+산성 | 2 |
| 중성+산성+알칼리성 | 3 |
| 알칼리성+중성 | 4 |
| 알칼리성+산성 | 5 |
| 병충해 | 응애 | 0 |
| 진딧물 | 1 |
| 깍지벌레 | 2 |
| 잎색 | 녹색, 연두색 | 0 |
| 녹색, 연두색, 흰계열 | 1 |
| 녹색, 연두색, 노란계열 | 2 |
| 녹색, 연두색, 붉은계열 | 3 |
| 녹색, 연두색, 은색계열 | 4 |
| 녹색, 연두색, 노란계열, 흰계열 | 5 |
| 녹색, 연두색, 흰계열, 붉은계열 | 6 |
| 붉은계열 | 7 |
| 기타 여러색 혼합 | 8 |
| 분류 | 잎보기 식물 | 0 |
| 잎+꽃보기 식물 | 1 |
| 잎+꽃+열매보기 식물 | 2 |
| 꽃보기 식물 | 3 |
| 잎+열매보기 식물 | 4 |
| 선인장 | 5 |
| 생육형태 | 풀모양 | 0 |
| 관목형 | 1 |
| 덩굴성 | 2 |
| 직립형 | 3 |
| 로제트형 | 4 |
| 생태형 | 일반형 | 0 |
| 수경형 | 1 |
| 건조형 | 2 |
| 잎무늬 | 무늬없음 | 0 |
| 줄무늬 | 1 |
| 점무늬 | 2 |
| 잎 가장자리 무늬 | 3 |
| 줄무늬+점무늬+가장자리무늬 | 4 |
| 꽃피는 계절 | 봄 | 0 |
| 여름 | 1 |
| 가을 | 2 |
| 겨울 | 3 |
| 봄, 여름 | 4 |
| 봄, 여름, 가을 | 5 |
| 봄, 겨울 | 6 |
| 여름, 가을 | 7 |
| 봄, 여름, 가을 | 8 |
| 봄, 여름, 가을, 겨울 | 9 |
| 꽃색 | 흰색 | 0 |
| 분홍계열 | 1 |
| 노랑 계열 | 2 |
| 분홍+노랑 계열 | 3 |
| 분홍+노랑+흰계열 | 5 |
| 분홍+노랑+흰+파랑계열 | 6 |
| 향기 | 거의없음 | 0 |
| 약함 | 1 |
| 중간 | 2 |
| 강함 | 3 |
| 번식방법 | 삽목 | 0 |
| 분주 | 1 |
| 파종 | 2 |
| 기타 | 3 |
| 배치장소 | 거실창측+발코니내측+발코니창측 | 0 |
| 거실내측+거실창측+발코니내측+발코니창측 | 1 |
| 어두운곳+거실내측+거실창측+발코니내측+발코니창측 | 2 |
| 발코니내측+발코니창측 | 3 |
| 거실창측+발코니내측 | 4 |
| 생장속도 | 빠름 | 0 |
| 보통 | 1 |
| 느림 | 2 |
| 겨울최저온도 | 0도이하 | 0 |
| 5도 | 1 |
| 7도 | 2 |
| 10도 | 3 |
| 13도이상 | 4 |
| 습도 | 40미만 | 0 |
| 40~70 | 1 |
| 70이상 | 2 |
| 건조에 강한 식물 | 배치장소 | 발코니 내측까지 이용 가능 | 0 |
| 발코니 창측까지 이용 가능 | 1 |
| 번식 | 잎꽂이 | 0 |
| 줄기 꽂이 | 1 |
| 파종 | 2 |
| 포기나누기 | 3 |
| 꺾꽂이 | 4 |
| 잎색 | 붉은색 | 0 |
| 녹색 | 1 |
| 검은색 | 2 |
| 혼합 | 3 |
| 자주색 | 4 |
| 꽃 | 흰색 | 0 |
| 노란색 | 1 |
| 붉은색 | 2 |
| 녹색 | 3 |
| 생장형 | 봄-가을 성장 | 0 |
| 가을-봄 성장 | 1 |
| 일년 내내 성장 | 2 |
| 관리수준 | 매우 쉬움 + 쉬움 | 0 |
| 보통 | 1 |
| 어려움+매우어려움 | 2 |
| 관리요구도 | 필요함 | 1 |
| 약간 돌봄 필요 | 2 |
| 잘 견딤 | 3 |
| 광요구도 | 낮은 광도 | 0 |
| 낮은 광도+중간 광도 | 1 |
| 중간 광도+높은 광도 | 3 |
| 높은 광도 | 4 |
| 생육적온 | 최고온도가 30도 이상일때 | 2 |
| 최고온도가 20 이상일 때 | 1 |
| 최고온도가 20 미만일 때 | 0 |
| 생태형 | 로제트형 | 0 |
| 군생형 | 1 |
| 관목형 | 2 |
| 불규칙형 | 3 |
| 탑형 | 4 |
| 직립형 | 5 |
| 포복형 | 6 |
| 직립형, 단측지 | 7 |
| 포도송이형 | 8 |
| 생장속도 | 느림 | 0 |
| 보통 | 1 |
| 빠름 | 2 |
| 뿌리형태 | 가는뿌리 | 1 |
| 굵은뿌리 | 0 |

[표 1]

구현 방법의 예시는 다음과 같다.



[그림 4]

**함수 구현**

**plant 함수**

plant함수는 사용자가 좋아요를 누른 식물의 번호를 입력값으로 받는다. 사용자 선호 식물의 벡터값을 리스트에 넣어 반환한다. 함수의 형태는 다음과 같다. 아래에서 설명할 contents\_based\_filtering 함수를 호출하여 최종 결과를 반환한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

[그림 5]

**contents\_based\_filtering 함수**

contents\_based\_filtering함수는 사용자가 좋아한 식물의 번호 리스트와 잎을 관상하는 식물과 건조 식물의 벡터 행렬을 입력값을 갖는다.

텍스트이(가) 표시된 사진

자동 생성된 설명

[그림 6]

해당 함수는 sklearn의 cosine\_similarity 메서드를 사용해 일차적으로 존재하는 모든 데이터의  코사인 유사도를 구한다. 이 때 잎 관상 식물은 잎 관상식물끼리, 건조식물은 건조식물끼리의 유사도 행렬을 구한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

[그림 7]

그 뒤로 입력값인 좋아요를 누른 식물이 잎을 관상하는 식물인가, 건조 식물인가에 따라서 유사도 행렬이 다르기 때문에 좋아요를 누른 식물의 종류를 먼저 확인한다. 좋아요를 누른 식물과 유사한 특성을 가진 (유사도가 높은) 식물을 추출한다. 이때 좋아요를 누른 식물 자신은 제외한 유사식물을 유사도 순으로 정렬한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

[그림 8]

그 뒤로 여러 좋아요 식물이 존재할 경우에 추천되는 식물이 겹칠 것을 고려하여 식물이 겹친다면 유사도가 높은 경우만을 고려하고 중복을 제거한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

[그림 9]

마지막으로 사용자에게 도출될 정보를 추출한다. 최종적으로 추천하는 식물의 정보와 유사도를 함께 출력하고, 반환값으로는 식물의 텍스트 정보를 제공한다. 예시로 실행시킨 함수의 출력된 식물의 번호와 유사도는 다음과 같다.

텍스트이(가) 표시된 사진

자동 생성된 설명

[그림 10]

중복으로 데이터가 겹치는 경우, 건조와 잎 관상 식물을 둘다 선택했을 경우에도 결과가 중복

없이 내림차순으로 정렬되는지 확인하기 위한 코드이다. 0번부터 215번까지는 잎 관상 식물, 216번부터 312번까지는 건조식물이다. a에는 잎 관상 추천 식물, b에는 건조 추천 식물이 할당된다.  출력되는 결과에서 튜플의 첫번째 인자는 식물번호, 두번째 인자는 유사도를 의미한다. 따라서 10개의 출력 결과 중 9개의 잎 관상 식물, 1개의 건조 식물이 추천되었다.

첫번째 인자를 기반으로 해당 인덱스를 갖는 식물들을 추출하여 데이터프레임을 새로 생성한다. 데이터프레임을 새로 생성하며 유사도를 새로운 칼럼으로 지정하여 추가한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

[그림11]

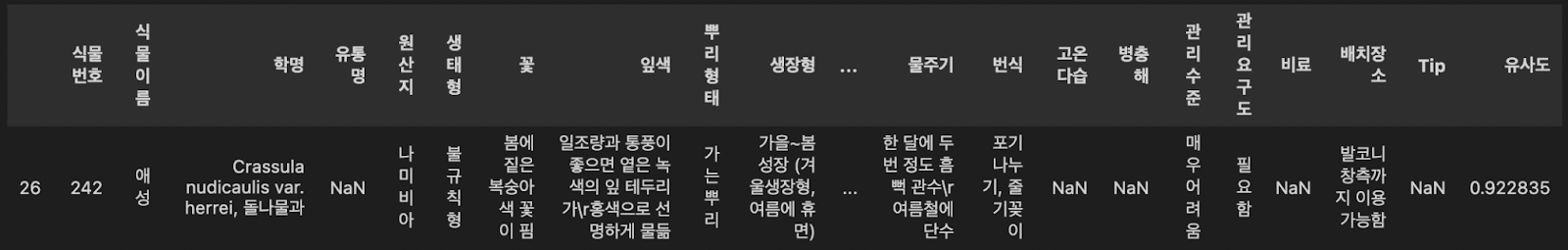
최종적으로 반환되는 인자는 다음과 같다. df1는 추천되는 잎 관상 식물의 텍스트 정보를, df2는 추천되는 건조 식물의 텍스트 정보이다.

[그림11]에서 설명한 것 처럼 저장된 a(추천 잎 관상 식물)를 출력한 결과는 다음과 같다. 해당 결과는 출력된 9개의 결과 중 3개의 결과이다.



[그림 12]

저장된 b(추천 잎 관상 식물)를 출력한 전체 결과는 다음과 같다.



[그림 13]

나. Flutter 애플리케이션

**plant/**

**텍스트이(가) 표시된 사진

자동 생성된 설명**[그림 13]

Plant 앱의 페이지 구성은 다음과 같다. 총 네 개의 화면으로 추천 피드를 보여 주는 home, 식물마다 상세 정보를 보여 주는 detail, 내가 좋아요를 누른 식물을 보여 주는 like, 내 정보를 수정할 수 있는 mypage로 구성돼 있다. 이후 앱을 실행하는 main.dart, BottomNavigationBar 클릭에 따라 페이지를 제어하는 root\_page.dart, 그리고 앱의 주된 색깔이나 UI 부분에서의 특징들을 상수로 선언해 놓은 constant.dart 파일이 있다.

**plant/constant.dart**

**텍스트이(가) 표시된 사진

자동 생성된 설명  
[**그림 14]

constant.dart 파일에는 포인트 색상으로 사용할 kPrimaryColor, 배경색으로 사용할 kBackgroundColor, 기본적인 패딩 간격인 kDefaultPadding을 미리 상수로 선언해 앱의 전반적인 UI의 통일성을 갖췄다.

**plant/main.dart**

**텍스트이(가) 표시된 사진

자동 생성된 설명**  
[그림 15]

앱이 처음 시작됐을 때 보이는 화면을 root\_page.dart 의 RootPage class로 설정하였다. 이후 RootPage 에서 BottomNavigationBar의 index 값에 따라 화면 제어를 하며 Scaffold를 바꿔 보여 준다.

**plant/root\_page.dart**

**텍스트이(가) 표시된 사진

자동 생성된 설명**[그림 16]

List 형식의 bodys에 식물을 추천해 주는 페이지의 바디 위젯인 HomeBody, 내가 좋아요를 누른 식물 목록을 보여 주는 페이지의 바디 위젯인 LikeBody, 마이페이지의 바디 위젯인 MypageBody를 요소로 설정하였다. 이후 RootPage에서 그리는 body가 클릭된 BottomNavigationItem의 index 값인 \_selectedIndex 번째의 bodys 요소가 페이지의 바디로 그려진다.

텍스트이(가) 표시된 사진

자동 생성된 설명  
[그림 17]

BottomNavigationBar가 클릭되었을 때 실행하는 \_onItemTapped 함수는 현재 BottonNavigationItem의 index 번호를 받아 \_selectedIndex에 저장하는 형식으로 구현했다. 위와 같은 root\_page를 만든 이유는 Navigation Bar 클릭에 따라 각각의 페이지로 이동하는 것이 가능하게 하기 위함이다.

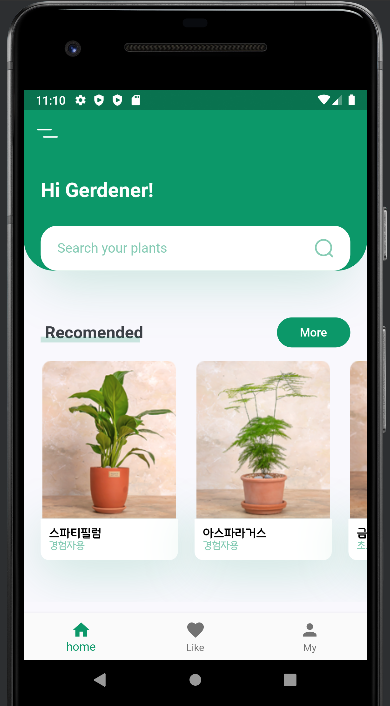
**plant/screens/home/**

**텍스트이(가) 표시된 사진

자동 생성된 설명**  
[그림 18]

home 폴더 안에는 사용자에게 알맞은 식물을 추천해 주는 페이지를 구현하기 위한 구성 요소들로 묶여져 있다.

**plant/screens/home/components/body.dart**

****  
[그림 19]

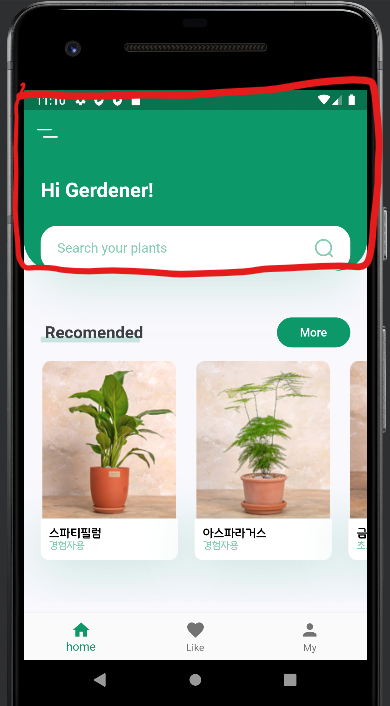
완성된 HomePage의 Body 모습은 위와 같다.

텍스트이(가) 표시된 사진

자동 생성된 설명  
[그림 20]

스크롤이 가능한 SingleChildScrollView를 반환하고, child로 HeaderWithSearch, TitleWithorBtn, RecommendPlants 클래스를 세로 방향인 2Column으로 묶어 가지고 있다.

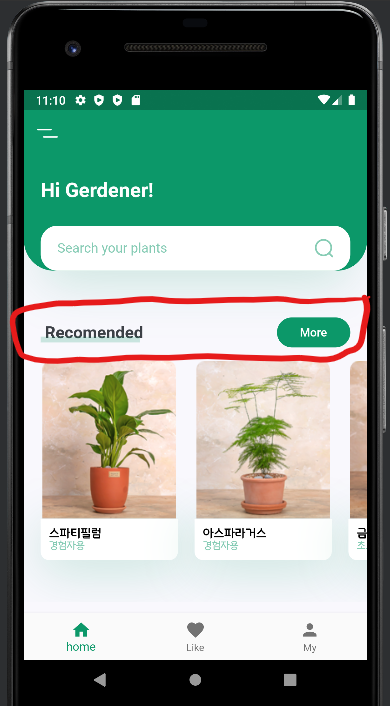
**plant/screens/home/comopnents/header\_with\_search.dart**

****

[그림 21]

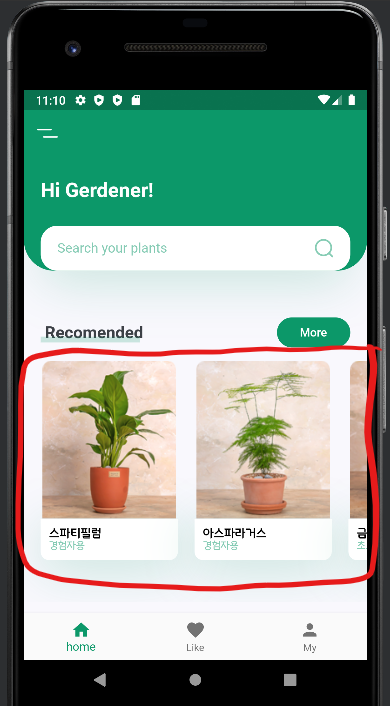
HeaderWithSearch 클래스는 위의 화면을 구현한 것이다.

**Plant/screens/home/components/title\_with\_more\_bbtn.dart**

****  
[그림 22]

TitleWithMoreBbtn 클래스는 위의 화면을 구현한 것이다.

**plant/screens/home/components/recomend\_plants.dart**

****  
[그림 23]

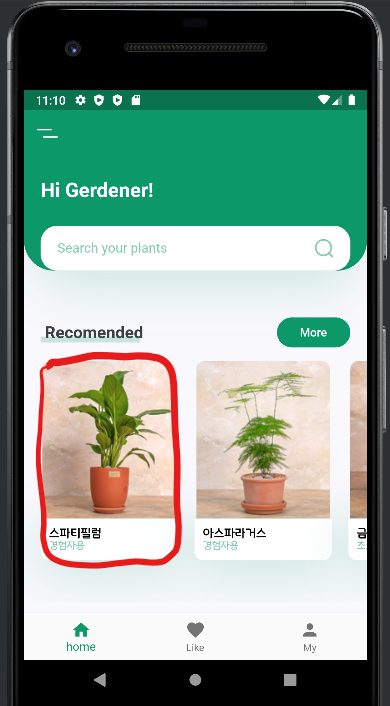
RecomendPlants 클래스는 위의 화면을 구현한 것이다.

텍스트이(가) 표시된 사진

자동 생성된 설명  
[그림 24]

SingleChildScrollView를 제공하며 스크롤 방향을 수평인 horizontal로 설정하였다. 그리고 각각의 식물 카드를 행인 Row로 엮어서 화면을 구성하였다. Row에 들어가는 식물 카드들은 GestureDetector 위젯을 사용해서 클릭이 감지되었을 때, Navigation을 통해 각각 식물의 세부 페이지로 이동할 수 있도록 하였다. Navigator에서는 MaterialPageRoute를 사용하였다. 또한 각각의 식물 카드들은 RecomenPlantCard\_one 등의 클래스로 반환해 Row를 구성하도록 하였다.

**plant/screens/home/components/recomend\_plant\_card.dart**

****  
[그림 25]

RecomendPlantCard 클래스는 다음 화면을 구현한 것이다.

텍스트이(가) 표시된 사진

자동 생성된 설명  
[그림 26]

BoxDecoration을 사용해 식물 카드의 겉부분이 동그랗고 매끄럽게 표현될 수 있도록 하였다. 또한 BoxShadow 기능을 이용해 카드 밑에 그림자를 넣어 디자인적인 요소를 챙겼다. 식물의 정보를 입력하는 TextSpan 부분에 Text의 다양한 성질을 변경하여 넣었다. 특히 fontFamily에서는 assets 파일에 font 파일을 넣고 pub get을 통해 직접 다운로드 받은 폰트를 적용하였다.

**plant/screen/detail**

**텍스트이(가) 표시된 사진

자동 생성된 설명**  
[그림 27]

detail 페이지는 식물마다 하나씩 페이지를 갖고, 페이지를 구성하는 components로 짜여 있다.

**plant/screens/details/detail\_screen\_one.dart**

**텍스트, 전자기기, 디스플레이, 컴퓨터이(가) 표시된 사진

자동 생성된 설명**[그림 28]

완성된 DetailPage의 모습은 다음과 같다.

텍스트이(가) 표시된 사진

자동 생성된 설명  
[그림 29]

DetailPage 의 Body를 구성하는 클래스에 어떤 식물의 세부 정보를 나타낼 것인지 입력값들을 parameter로 전달하여 페이지를 그리는 방식이다.

**plant/screens/detail/components/body.dart**

**텍스트이(가) 표시된 사진

자동 생성된 설명**  
[그림 30]

전달받은 parameter 중 image는 ImageandIcon 클래스로 넘기고, 나머지 parameter들은 DrtailInfo 클래스로 넘긴 뒤, 두 클래스를 Column으로 엮어서 화면을 구성한다. SingleChildScrollView로 구성하였고, 수직으로 스크롤 가능하도록 방향을 vertical로 설정하였다.

**plant/screens/detail/components/image\_and\_icons.dart**

**텍스트, 주방기기이(가) 표시된 사진

자동 생성된 설명**  
[그림 31]

ImageAndIcons 클래스는 위 화면을 구현한 것이다.

텍스트이(가) 표시된 사진

자동 생성된 설명  
[그림 32]

네 개의 icon들을 수직으로 엮고, 오른쪽엔 인자로 받은 image를 배치하였다. 첫 번째 icon인 좋아요 버튼은 눌렀을 때 \_hasBeenPressed가 참이라면 red, 거짓이라면 black으로 나타나게 하였다.

텍스트이(가) 표시된 사진

자동 생성된 설명[그림 33]

setState 함수를 사용하여 좋아요 버튼을 눌렀을 때 \_hasBeenPressed 값을 true에서 false로, false에서 true로 바꾸도록 하였다.

**plant/screens/detail/components/detail\_info.dart**

**텍스트, 모니터, 실내, 전자기기이(가) 표시된 사진

자동 생성된 설명**  
[그림 34]

DetailInfo 클래스의 구현은 위와 같다.

**plant/screens/like/**

LikePage는 HomePage를 응용하여 만들었다.

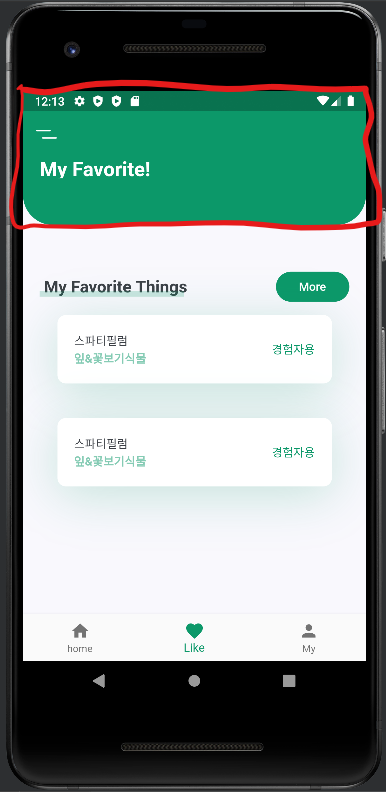
**plant/screens/components/body.dart**

**텍스트, 모니터, 전화, 휴대폰이(가) 표시된 사진

자동 생성된 설명**  
[그림 35]

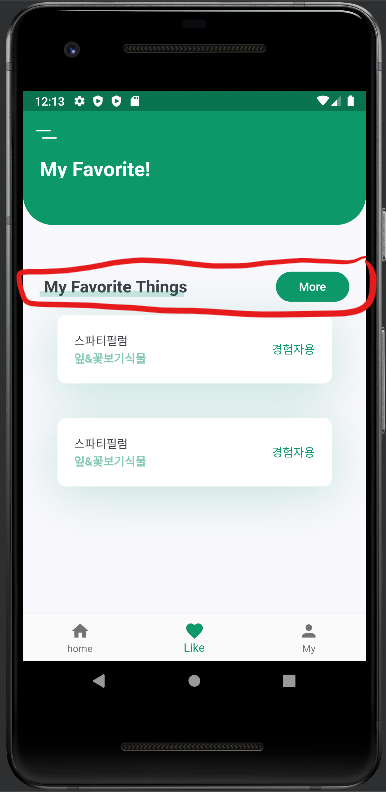
완성된 LikePage의 body는 다음과 같다.

**plant/screens/like/components/header.dart**

****  
[그림 36]

HomePage를 만들 때 사용했던 HeaderWithSearch 클래스의 사이즈를 줄이고 검색창을 없앴다.

**plant/screens/like/components/title\_with\_more\_bbtn.dart**

****  
[그림 37]

HomPage의 TitleWithMoreBbtn 클래스를 그대로 활용하여 텍스트만 변경하였다.

**plant/screens/like/favorite\_plants.dart**

**텍스트, 모니터, 휴대폰, 전화이(가) 표시된 사진

자동 생성된 설명**  
[그림 38]

HomePage의 RecomendPlants 클래스를 활용했다. 카드를 가로 스크롤 배치에서 세로 스크롤 배치로 변경하였다. 마찬가지로 해당 카드들을 클릭했을 때 식물의 상세 페이지로 이동할 수 있도록 Navigation을 추가하였다.

**plant/screens/like/components/favorite\_plant\_card.dart**

**텍스트, 모니터, 전화, 휴대폰이(가) 표시된 사진

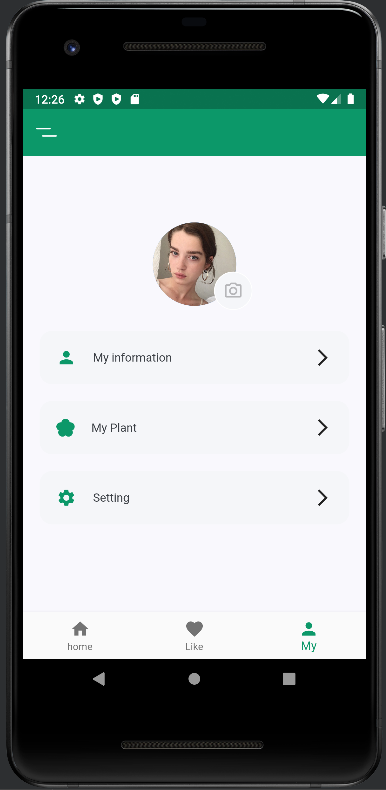
자동 생성된 설명**  
[그림 39]

HomePage의 RecomendPlantCard 클래스를 활용하였다. 카드의 크기를 세로로 길게 바꾸고, 상단에 위치하던 이미지를 제거하였다.

**plant/screens/mypage/**

mypage는 body를 return하는 …/mypage/body.dart 파일로 구성돼 있다.

**plant/screens/mypage/body.dart**

****  
[그림 40]

완성된 mypage의 body 부분은 다음과 같이 구현된다.

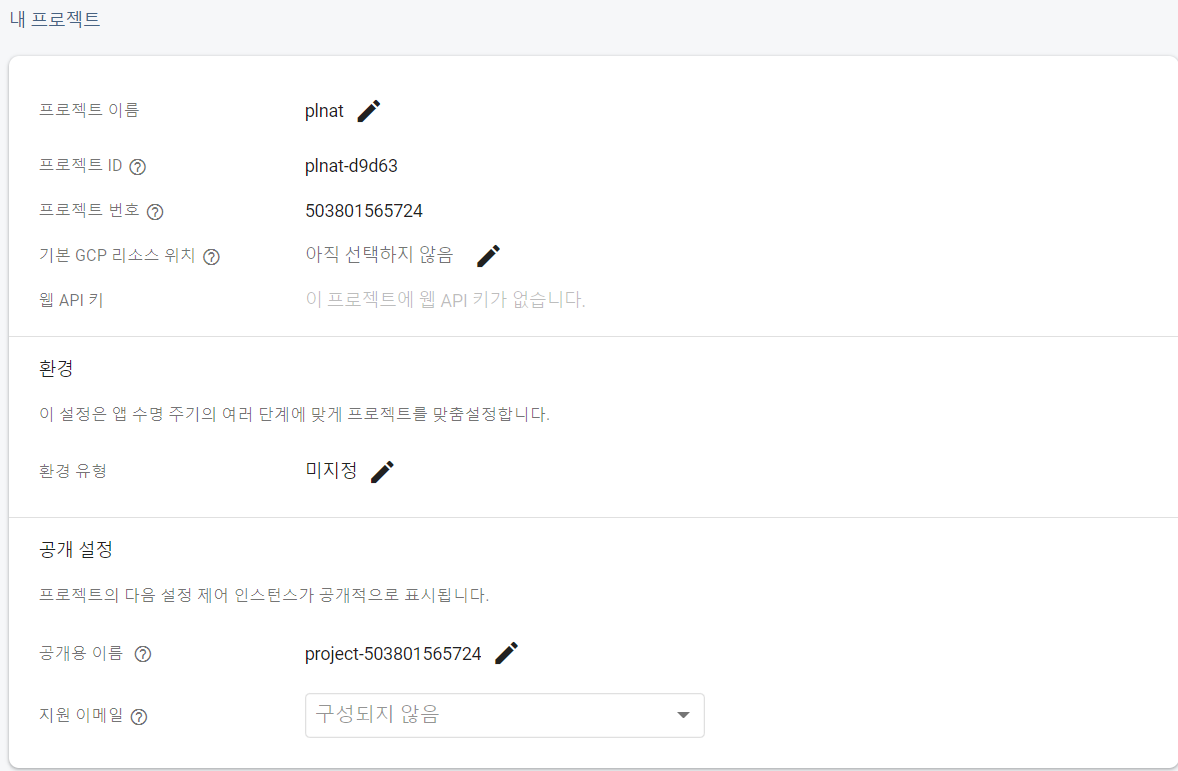
텍스트이(가) 표시된 사진

자동 생성된 설명  
[그림 41]

Mypage는 사용자의 프로필 밑에 padding으로 크기를 지정한 FlatButton을 세로 방향인 Column으로 묶어 배치하였다. 그리고 각각의 FlatButton 안에는 asset으로 추가한 icon, text, 기본 화살표 icon을 가로 방향인 Row로 묶어 삽입하였다.

**FireBase**

**Google firebase**

****[그림 42]

구글에서 제공하는 firebase에서 프로젝트를 생성해 데이터를 연동하였다. Firebase 프로젝트를 생성하고, 안드로이드용으로 만들어 json 파일을 다운로드하였다.

**Android studio setting**

**텍스트이(가) 표시된 사진

자동 생성된 설명**[그림 43]

이후 다운받은 json 파일을 plant/android/app/ 폴더에 삽입해 주었다.

텍스트이(가) 표시된 사진

자동 생성된 설명[그림 44]

이후 plant/android/build.gradle 파일에 플러그인을 설치해 주었다.

텍스트이(가) 표시된 사진

자동 생성된 설명[그림 45]

plant/android/app/build.gradle 파일에도 파이어베이스 path를 설정하였다.

**제 3 장 성과요약 및 기대효과**

제 1 절 핵심 개발성과 및 독창적 핵심기술

본 프로젝트의 핵심적인 개발성과는 딥러닝 및 추천 시스템 알고리즘으로 사용자들에게 보다 정량적이며 객관적인 모델링 결과에서 비롯된 식물 콘텐츠를 추천해주는 것이다. 기존의 서비스나 정보 사이트에서 식물을 추천하는 메커니즘은 단순히 정보 나열, 카테고리형 추천이 대부분이었고, 소비자가 선호도나 특성에 따른 레이블을 체크하면 그와 맞는 식물을 매칭시켜주는 방식도 존재했다. 따라서 카테고리에 따른 정보 나열이나, 단순 매칭의 추천이 아닌 유사도 기반 추천 알고리즘으로 본 프로젝트만의 독창적인 식물 추천 기술을 구현하였으며, 결론적으로 사용자가 선택한 다양한 식물을 기반으로 보다 맞춤화된 정보를 제공해줄 수 있을 것으로 기대된다.

제 2 절 기대효과

본 프로젝트의 기대 효과는 다음과 같다.

첫째로 사용자는 번거로운 데이터 입력 과정이나 정보 탐색 과정 없이도 자신에게 알맞는 식물을 추천 받을 수 있고 여러 추천 정보들을 소비자가 직접 비교하며 본인에게 잘 맞는 식물을 다양하게 선택할 수 있는 장점이있다.

두번째로 소비자가 선택한 식물들을 기반으로 단순하게 매칭 시켜주는 것이 아닌 코사인 유사도를 가지고 있는 식물 DB별로 비교를 통해 소비자가 선택한 다양한 식물들과 비교를 통해 최적의 코사인 유사도를 가진 식물을 정량적인 평가를 통해 소비자에게 객관적으로 보여줄 수 있다는 장점이 존재한다.

세번째로 해당 기술을 기반으로 현재 본 연구에서는 식물 종류에 있어 한정된 DB를 가지고 있지만, 해당 기술을 기반으로 더욱 다양한 카테고리별 식물 DB를 구축하고 본격적인 서비스화가 이루어진다면 더욱 알찬 콘텐츠 구성 및 추천을 가능하게 할 것이다.

제 3 절 마케팅 전략 및 향후 추진 계획

해당 서비스가 식물 어플 도메인에 나온다면 해당 서비스를 통해서 소비자들은 기존의 서비스보다 더 객관적이고 정량적인 추천을 통해서 본인에게 잘맞는 다양한 종류의 식물을 추천받을 수 있을 것이다. 서비스에 대한 호응도가 좋다면 해당 어플에 담기는 식물 데이터셋도 더 증가할 것이며 더욱 다양한 식물 카테고리별 추천이 가능해지는 형태로 발전해 나갈 것이다. 그에따라 추가적인 서비스로 추천한 식물들을 구매할 수 있는 추가적인 기능을 구현하거나 리뷰 시스템 같은 기능을 추가함으로써 고객의 제품 선택에 더 다양한 수단을 통해 결정할 수 있도록 도울 것이다. 리뷰 시스템 같은 경우에도 리뷰수가 많아진다면 소비자가 남긴 리뷰를 통해 분석을 진행해서 해당 식물을 구매 및 실제로 기른 사람들이 남긴 리뷰를 기반으로 리뷰 분석을 진행해서 실제로 식물에 나와있는 특징 외에도 소비자가 느끼는 특징 같은 정보들도 추가적으로 제공함으로써 서비스의 품질을 향상시킬 수 있을 것이다. 이 외에도 다양한 서비스화를 통해서 소비자가 보다 식물을 추천 받는데 도움을 받을 수 있는 다양한 서비스를 구현할 것이며 발전시켜나갈 계획이다. 또한 현재 검색 기능이 UI로만 구현돼 있고 실제 식물을 검색할 수 있는 기능적인 부분이 구현되지 않았다. 추후 데이터 분석을 통해 정리한 데이터들을 최대한 활용할 수 있도록, 플러터 애플리케이션에서 시각화할 수 있는 구현 능력을 갖춰 해당 기능이 작동하도록 할 것이다.

**제 4 장 소스 코드 출처 및 참고문헌**

제 1 절 소스 코드 출처 및 사용범위

[1] <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html>

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

sklearn에서 제공하는 소스코드 중에서 cosine\_similarity 메소드를 사용하였다. 해당 소스코드는 데이터 행렬에 대한 유사도를 구하며, 반환값으로 유사도 정보를 담은 행렬을 반환한다. cosine\_similarity 메소드를 사용해 잎 관상 식물과 건조식물 각각의 유사도를 구한 뒤, 반환된 유사도 행렬을 기반으로 추천식물을 도출하였다.

[2] github: iampawan/Cocktail-App <https://github.com/iampawan/Cocktail-App>

[3] github: firebase/flutterfire <https://github.com/firebase/flutterfire>

[4] github: JideGuru/FlutterFoodybite <https://github.com/JideGuru/FlutterFoodybite>

제 2 절 참고 문헌

[1] 코로나 블루에 ‘반려식물’ 찾는 사람들 늘었다, 서울경제, 2020.10.03

[2] 마크로밀 엠브레인, 식물 관리에 대한 인식 조사, 2017

[3] 네이버 표준 국어대사전, 2019

[5] 박승택, 「사이버커뮤니케이션학보, 통권 제34 권 1호」, 『기계학습 기반의 뉴스 추천 서비스 구조와 그 효과에 대한 고찰\*: 카카오의 루빅스를 중심으로』,, pp. 18-27, 2017

[6] Data 제공 농사 관련 공공데이터 포털 ([농사로 농업기술포털 (nongsaro.go.kr)](https://www.nongsaro.go.kr/portal/portalMain.ps?menuId=PS00001))

[7] 김예나, 최인복, 박태근, 이재동. (2008). 사용자 기반과 아이템 기반 협업여과 추천기법에 관한 실증적 연구. 한국정보처리학회 학술대회논문집, 15(2), 714-717.

[8] 유송은, 강병오, 김지민, 이강혁, 이민우, 고석주.(2020).코사인 유사도 분석을 이용한 최저가 매칭 서비스.한국방송미디어공학회 학술발표대회 논문집,(),502-507.

[9] 홍동균, 이연창, 김상욱, and 이종욱. "추천 시스템에서의 컨텐츠 기반 필터링 기법의 효과." 한국정보과학회 학술발표논문집 2017.6 (2017): 266-67. Web.

개인별 기여 사항 (1인 구성팀은 아래 본인의 기여 사항만 적으면 됩니다.)

|  |  |
| --- | --- |
| 성명 | 기여 내용 (기여내용은 개인당 최대 1페이지 분량으로 작성) |
| 조희수 | 1. 크롤링을 이용한 데이터 수집: 셀레니움을 이용해 농사로 사이트에서 식물에 대한 정보를 크롤링했다. 농사로의 실내식물 정보에서 잎을 관상하는 실내식물의 상세 페이지로 넘어가는 링크를 크롤링하고, 셀레니움으로 해당 링크에 접속해 식물의 정보를 수집하였다. 2. Proposal 작성: 기대효과 부분 작성에 기여하고 전반적인 검토와 피드백 과정을 함께하였다. 또한 ER Diagram와 IA를 draw.io로 그려서 보고서에 삽입하였다. 3. Flutter 애플리케이션 개발: Plant 애플리케이션을 개발하였다. 페이지들을 제어하는 RootPage, 식물을 추천하는 페이지인 HomePage, 내가 좋아요 버튼을 누른 식물만 보여 주는 LikePage, 식물의 상세 정보를 보여 주는 DetailPage를 구현하였다. 이외에도 BottomNavigationBar를 이용한 화면 제어와 애플리케이션 기능 내부에서 일어나는 모든 Navigator 이동, onState 함수를 통한 위젯 그리기를 구현하였다. 4. FireBase 연동: google의 firebase를 이용해 프론트 엔드와 백 엔드의 정보 통신이 용이하게 진행될 수 있도록 Flutter 애플리케이션에 apply를 해 주었다. 5. 최종보고서 작성: 제 2절 세부 개발내용 및 방법 중 가-2 Flutter 애플리케이션 기술 개발 추진 방법 및 전략, 나-2 Flutter 애플리케이션 기술개발 주요 메뉴 구조, 다-3 Flutter 개발 환경, 라-2 Flutter 애플리케이션 개발 부분을 작성하였다. 6. 최종 발표: Flutter 애플리케이션 개발 환경 슬라이드부터 Flutter 애플리케이션과 관련된 발표 내용을 맡았고, 이후 성과 요약과 기대 효과 부분까지 발표를 맡았다. |

|  |  |
| --- | --- |
| 성명 | 기여 내용 (기여내용은 개인당 최대 1페이지 분량으로 작성) |
| 이수진 | 1. 크롤링을 통한 데이터 수집 : 농사로 홈페이지에서 식물 특성 및 정보에 대한 데이터를 크롤링했다. 실내 식물 정보에서 건조식물의 상세 페이지로 넘어가는 링크를 크롤링하고, 태그검색으로 상세 정보를 수집해 엑셀파일로 변환했다. 또한 식물 정보에 해당하는 식물 사진을 수집하여 저장하였다. 이 과정에서 잎 관상 식물과 건조 식물의 데이터를 취합하여 전체적으로 수정하는 과정을 거쳤다. 2. Proposal 작성: 컨텐츠 기반 필터링 알고리즘의 구체적인 구조와 실행 단계에 대해 서술하였다. 3. Proposal PPT 제작: Proposal 내용을 기반으로 PPT를 제작하였다. 4. 데이터 전처리 : 전체 데이터의 칼럼 이름을 수정하고, 필요한 칼럼만을 추출하여 csv 파일로 제작하였다. 그 이후 잎 관상 식물 데이터의 카테고리화를 진행하고, 결측값 처리를 진행하였다 . 5. 컨텐츠 기반 필터링 알고리즘 제작 : 식물 데이터를 입력 받으면 코사인 유사도를 구해 유사도가 높은 10개의 추천 식물을 중복을 제거하고 추출하는 contents\_based\_filtering 함수를 제작하였다. 이 과정에서 최종적으로 추출되는 정보를 CSV 파일로 저장하도록 코드를 구현하였다. 6. 최종 보고서 작성 : 제 2절 개발 내용 중 모델에 대한 개발 추진방법 및 전략, 주요 개발 내용 및 구조, 개발 환경, 개발 내용에 대한 보고서 작성을 맡았다. 7. 최종 보고서 PPT 제작 : 컨텐츠 기반 필터링 모델 구현 과정 전체 및 결론 기대효과 부분의 PPT 제작을 맡았다. |

|  |  |
| --- | --- |
| 성명 | 기여 내용(기여내용은 개인당 최대 1페이지 분량으로 작성) |
| 노희지 | 1. 데이터 수집: 농사로 사이트의 각 카테고리의 페이지 링크를 크롤링 하는 작업을 진행하였다 2. Proposal: business understanding 파트를 담당하여 proposal을 작성하였다. 3. IA 제작: 앱의 정보구조도인 IA를 제작하였다 4. 와이어프레임 제작: 앱 화면 구성에 대한 와이어프레임과 해당 와이어프레임을 이용해 플로우 차트를 제작하였다. 5. Ui 틀 제작: 본격적인 개발 작업 전 피그마를 이용해 와이어프레임에서 한층 발전된 ui를 제작하였다. 6. Flutter 앱 구현: 앱 내에서 사용자가 자신의 정보를 비롯해 식물 정보를 관리할 수 있는 my 뷰를 구현하였다. 7. 더미데이터 수집: 앱 내에서 필요한 더미데이터를 수집 및 정리하였다. 8. Ppt 제작: ppt의 기본적인 템플릿을 비롯해 1장 개발기술의 개요와 3장 플러터 앱 구현 파트의 PPT를 제작하였다. 9. 최종 보고서: 1절 개발기술 개요의 전체 부분, 2절 기술 개발 목표 중 앱 파트 내용, 3절 개발 목표 및 개발 내용 중 기술개발 주요 메뉴 구조 파트에 대한 보고서를 작성하였다. 10. 보고서 최종 취합 |

|  |  |
| --- | --- |
| 성명 | 기여 내용(기여내용은 개인당 최대 1페이지 분량으로 작성) |
| 문선웅 | 1. Proposal 작성: 모델링 파트인 컨텐츠 기반 필터링 알고리즘의 구체적인 구조와 실행 단계에 대해 서술하였다. 해당 연구가 기존 어플들과 보이는 차이점 및 개요 작성도 진행했다. 2. proposal 발표 및 영상 녹화: proposal 발표를 담당하고 녹화를 진행했음. 3. 모델링 데이터 전처리를 진행했다. 건조식물 DB를 구축하고 각 카테고리 별 레이블링 작업을 진행했다. 4. 컨텐츠 기반 알고리즘 소스코드 서칭 및 flow chart 구현을 시도했다. 참조코드를 기반으로 컨텐츠 기반 알고리즘 구현을 시도했다. 5. 보고서 작성 : 2장 1절 최종목표, 기능 기술개발 목표, 기술개발 변경사항, 3장 1절 핵심 개발성과 및 독창성, 2절 기대효과 작성을 담당했음 6. 최종 발표 영상 녹화: 개요 파트 및 모델링 파트 발표 및 녹음을 담당했음. |

**팀 자체 평가**

|  |  |  |
| --- | --- | --- |
| 항목 | 점수 | 사유 |
| 충실성  (20) | 20 | 1. 매주 2회씩 회의를 진행 2. 구글 공유문서를 통해 수정을 자주 진행 3. 최종 보고서에 들어가야 할 내용을 충실히 조사 4. 각자 맡은 역할을 충실히 이행 |
| 구체성  (20) | 20 | 1. 모델링을 보다 구체적으로 하기위해 ML Model 구성에 힘씀 2. 컨텐츠 기반 필터링 알고리즘의 로직을 구체적으로 구현하였음 3. 최종적으로 반환되는 추천식물의 특성을 구체적으로 정의함 4. 플로우차트, ERDiagram을 통해 서비스를 구현하였을 때 어떤 방식으로 서비스가 구동되는지 구체화함 5. IA와 와이어프레임에 기반해 각 화면들을 구현하였고, 앱의 디테일을 살리고자 다양한 설정값들을 활용 6. 기능마다 화면을 나누고 Navigator를 이용해 페이지 간 이동을 제어함 7. 데이터 분석 모델을 사용자가 쉽게 이용할 수 있도록 애플리케이션에 구현함 8. Color, font, padding 등 디자인적 요소를 통합하여 가독성 좋은 화면을 제공함 |
| 타당성  (20) | 15 | 1. 카테고리에 따른 정보 나열이나, 단순 매칭의 추천이 아닌 유사도 기반 추천 알고리즘으로 본 프로젝트만의 독창적인 식물 추천 기술을 구현 2. 사용자가 선택한 다양한 식물을 기반으로 보다 맞춤화된 정보를 제공 |
| 실현가능성  (20) | 15 | 1. 구체적인 모델의 구성과 방법, 다른 서비스에서 사용한 예시가 존재하기 때문에 충분히 실현 가능 2. 실제 사용자의 피드백을 기반으로 서비스를 구체화 시키지 못한다는 점에서 어려움을 겪을 수 있음 3. 데이터의 개수가 한정적이라는 도메인의 특성으로 보아 구현하는데 어려움을 겪을 것으로 예상 |
| 사업성  (20) | 20 | 1. 기존의 서비스들은 카테고리 형식 추천, 사용자 직접 입력 기반 추천, 단순 매칭의 방식의 수준에서 제품을 추천해주는 실상이기 때문에 정량적이고 객관적인 지표인 ML 작업을 거친 해당 프로젝트의 결과물은 충분히 사업성이 있을 것으로 예상 2. 비즈니스 배경에서 언급했듯 최근 코로나 19로 인해 반려식물에 대한 관심이 급증하며 실제 반려식물을 비롯해 관련 조경 물품의 판매량이 늘어나고 있음 따라서 해당 결과물은 이러한 반려식물에 많은 관심을 가지고 있는 사람들의 니즈를 공략할 수 있을 것임 |