

팀 프로젝트 최종 보고서

소비 행동 패턴 기반 나만의 AI 금융 코치

Team Moni

LGU+ Why Not SW Camp 8기

2025.01.07

팀장 : 박시하

팀원 : 김단하, 박소현, 홍예은

목 차

1. 요약문.....	2
2. 서론.....	2
2.1 추진 배경 및 문제 정의.....	2
2.2 프로젝트 목표.....	2
3. 관련 연구 및 서비스 분석.....	3
3.1 기존 서비스/연구 조사.....	3
3.2 차별점.....	3
4. 설계 및 구현.....	4
4.1 개발 환경.....	4
4.2 시스템 아키텍처.....	4
4.3 데이터/DB 설계.....	5
4.4 API 설계.....	7
5. 기능 설계 및 구현 상세.....	8
5.1 기능 1: 가상 마이데이터 생성.....	8
5.2 기능 2: 소비 유형 분류.....	9
5.3 기능 3: 이상소비 탐지.....	11
5.4 기능 4: 머신러닝을 통한 소비 예측.....	12
6. 구현 결과.....	13
6.1 화면(UI) 구현 결과.....	13
6.2 배포/운영.....	18
7. 개인별 작업 내용.....	19
7.1 박시하.....	19
7.2 김단하.....	21
7.3 박소현.....	23
7.4 홍예은.....	26
8. 결론 및 회고.....	28
8.1 문제점 분석.....	28
8.2 향후 계획.....	28
8.3 기대 효과.....	29
9. 참고문헌.....	30

1. 요약문

본 프로젝트는 예산 초과, 충동 소비, 저축률 감소 등 금융문제를 인지하지 못하는 사용자의 소비 행동 데이터를 분석하여 패턴 변화를 사전에 감지하기 위해 시작되었다. 따라서 사용자 소비 행동 데이터와 AI 분석을 결합하여 유형분석 및 예측 알림기능을 제공하고, 핵심 기능은 소비 유형 자동 분류 기반 저축률 증가 유도, 이상소비 탐지, 머신러닝을 통한 소비 예측까지 총 3가지로 구성된다. 최종적으로 실제 서비스 적용 가능성을 고려한 프로토타입 수준의 시스템을 구현하였다.

2. 서론

2.1 추진 배경 및 문제 정의

소비자는 일상적인 소비 활동 과정에서 충동 소비나 예산 초과와 같은 문제를 반복적으로 경험하고, 이러한 소비 행태가 저축률 감소로 이어지고 있음을 인지하지 못하는 경우가 많다. 기존 금융 분석 및 가계부 서비스는 소비 내역의 조회와 단순 통계 제공에 집중되어 있으며, 예산 초과나 이상 소비 발생 이후에 사후 알림을 제공하는 한계를 가진다. 이로 인해 사용자는 자신의 비합리적인 소비가 발생한 이후에야 인식하게 된다.

2.2 프로젝트 목표

본 프로젝트의 목표는 사용자의 일·주 단위의 소비 분석을 통해 미래 사전 예측 및 이상 소비 실시간 감지 알림을 보내어 계획적인 소비를 유도하고, 소비 행동 데이터를 분석하여 소비 유형을 진단하여 저축률 증가를 유도하는 것이다. 또한 금융 마이데이터를 기반으로 소비 내역 조회, 가계부 기록, 저축 목표 관리 기능을 제공함으로써 사용자가 자신의 소비 습관을 체계적으로 관리할 수 있도록 하는 것을 목표로 한다.

2.3 페르소나 정의

- 20~30대 사회초년생
- 실제 지출 관리가 어려워 저축으로 이어지지 못하는 직장인
- 재무관리 미숙 + 예산관리 의지가 있는 사용자

3. 관련 연구 및 서비스 분석

3.1 기존 서비스/연구 조사

현재 금융 시장의 대표적인 플랫폼의 핵심 기능은 다음과 같다.

플랫폼	핵심 기능	부족한 점
Toss	송금 → 종합 금융	소비 행동 패턴 기반 코칭 기능 부재, 소비 습관 교정 기능 미흡
Banksalad	데이터 기반 금융 상품 추천	실시간 피드백 및 행동 교정 기능 부재
KakaoPay	결제 → 자산 관리	소비 행동 패턴 변화 감지 및 교정 기능 부재
네이버 금융	결제·포인트 → 생활 밀착 소비	재무 습관 교정 기능 부재, 소비 행동 패턴 분석 미흡

3.2 차별점

예산 초과, 충동 소비, 저축률 감소와 같은 반복적인 금융 문제가 지속적으로 발생하고 있음을 확인하였고, 이를 보완하기 위해 아래와 같은 기능을 제공하고자 한다.

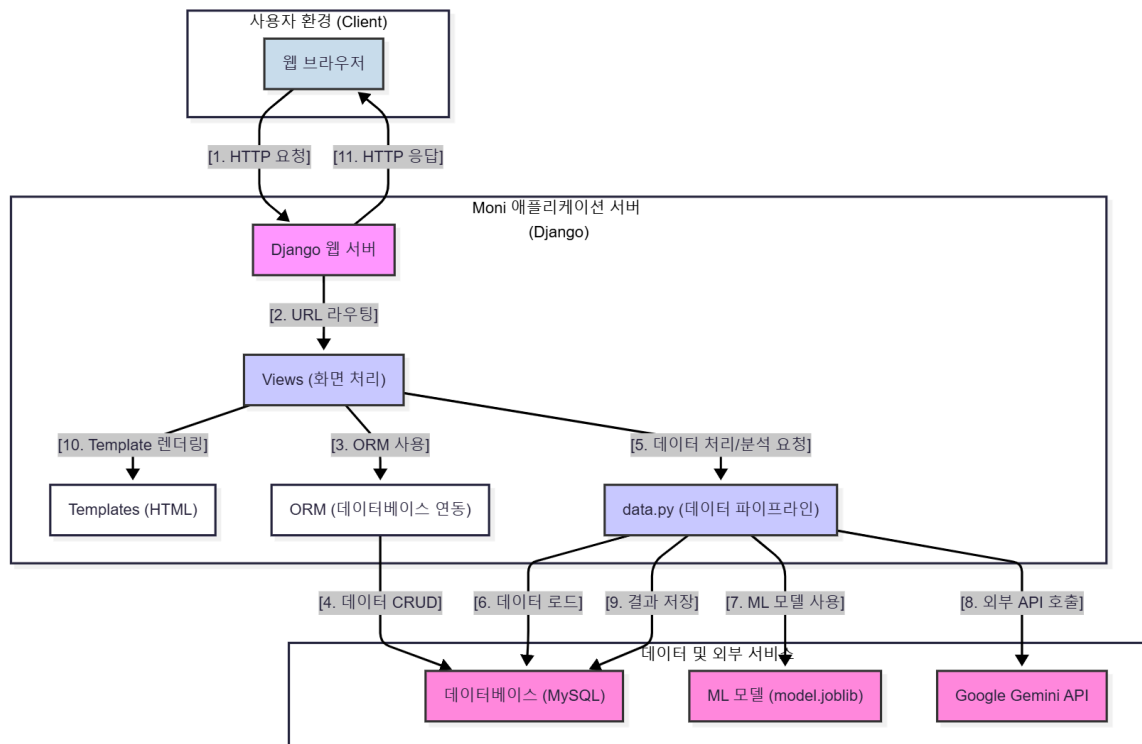
- 이상 소비 실시간 감지 알림
- 예산초과 가능성 사전 예측 & 사전 알림
- 소비 유형 자동 분류 기반 저축률 증가 유도
- 카드사별 혜택 제휴 고도화
- 통신 데이터 기반 초개인화 AI 코칭 (ixi-MO)
- 금융 상품 추천 고도화

4. 설계 및 구현

4.1 개발 환경

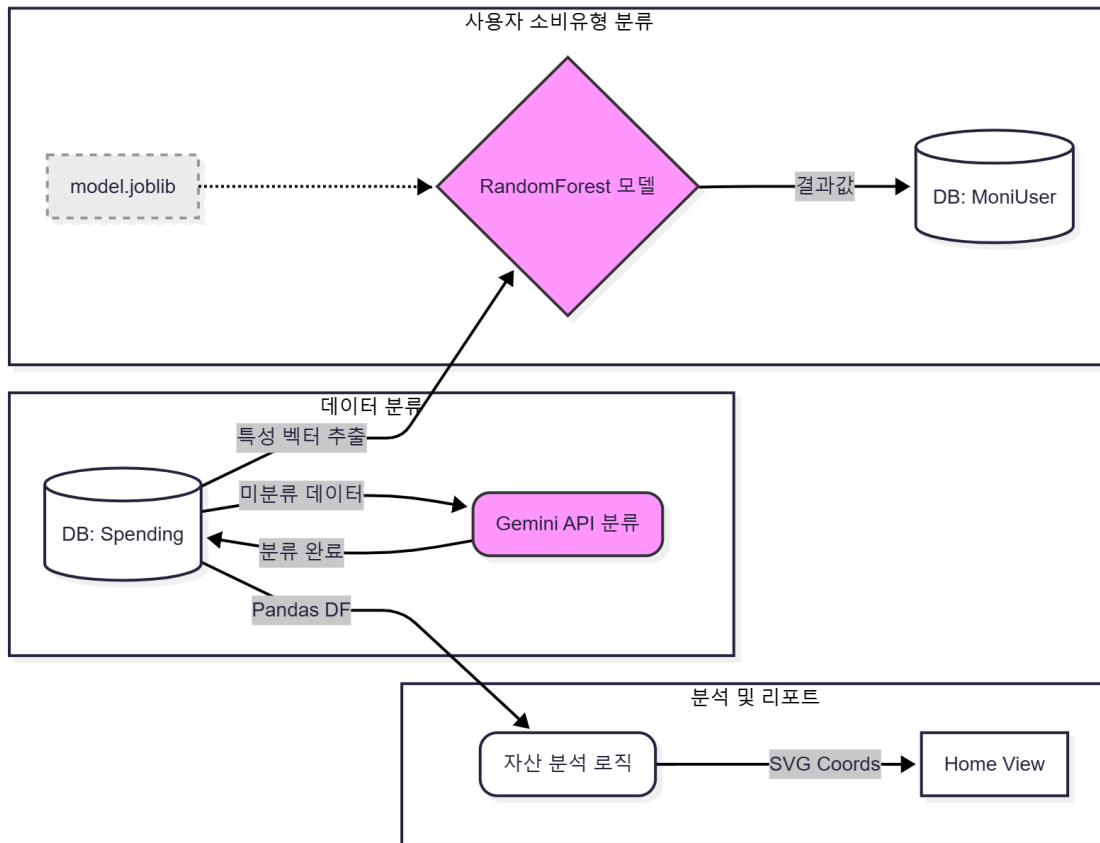
- Frontend: [Django/CSS, Java Scripts, html/Visual Studio Code 1.10]
- Backend: [Django, Requests/Python/Visual Studio Code 1.107]
- Database: [9.0.1 MySQL Version/AWS RDS]
- Infra/Hosting: [AWS EC2/Windows/Gunicorn]
- VCS/Collaboration: [Git/GitHub/Notion]

4.2 시스템 아키텍처



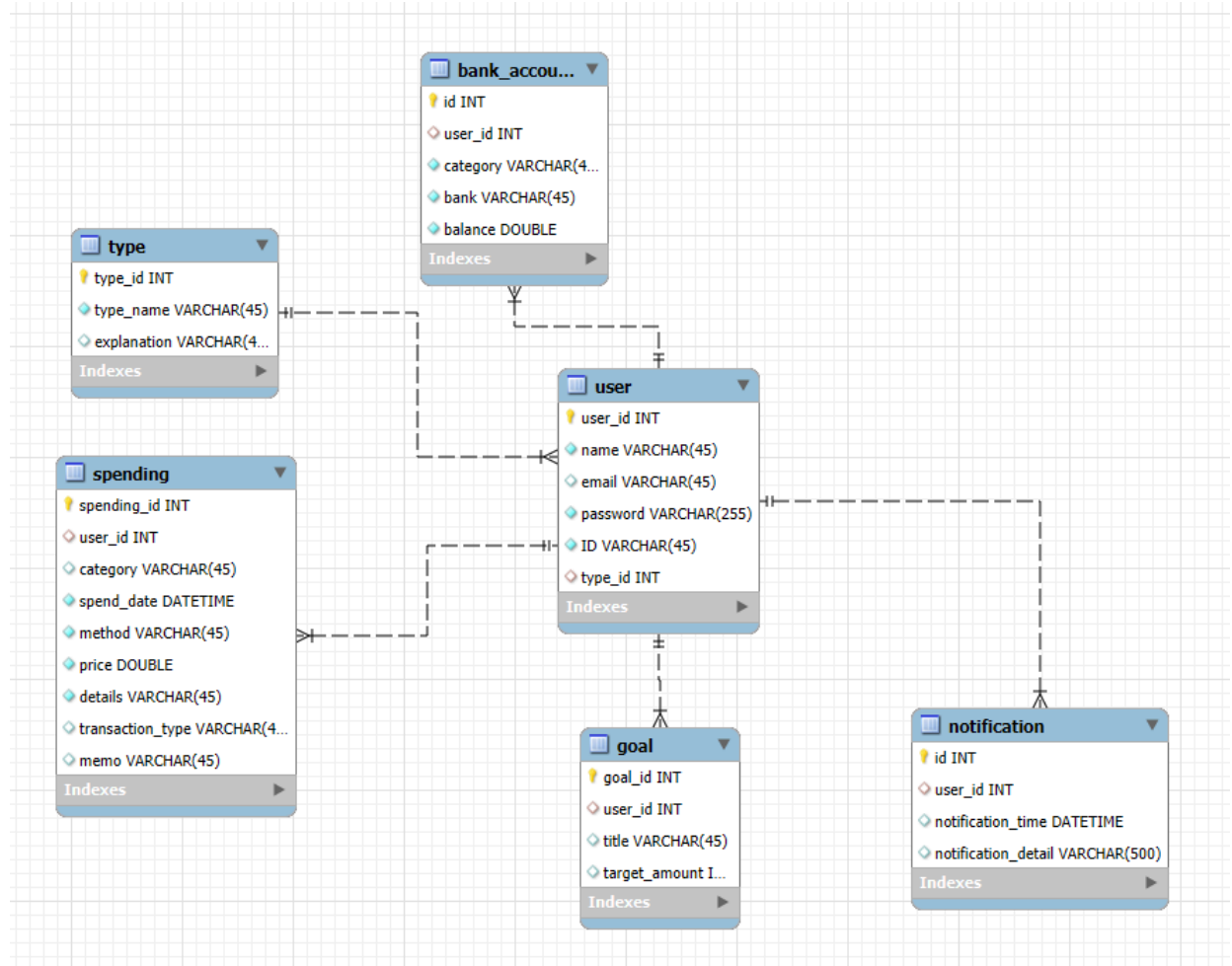
4.3 데이터/DB 설계

- 데이터 흐름도



- 데이터 전처리 및 분류
 - 지출 데이터를 **Gemini API**로 보내 미분류 항목을 자동으로 카테고리화
 - 분류된 데이터를 **Pandas DataFrame**으로 변환하여 분석 준비
 - 머신러닝 기반 소비유형 분석
 - 정제된 데이터에서 특성 추출
 - 저장된 **RandomForest** 모델을 로드해 사용자의 소비 유형 예측하여 결과를 사용자 **DB**에 저장
- 결과 시각화
 - 분석 로직을 거쳐 차트 생성을 위한 **SVG** 좌표 계산
 - 최종 분석 리포트를 사용자의 메인 화면에 시각화

• ERD



• 주요 엔티티:

- [User, Spending, Type, Goal, Bank_account, Notification]

• 관계

User - Spending : 사용자는 여러 개의 소비 내역을 가질 수 있는 **1:N 관계**이다.

User - Bank Account : 사용자는 여러 개의 계좌를 가질 수 있는 **1:N 관계**이다.

User - Notification : 사용자별 알림을 관리하기 위해 **1:N 관계**이다.

User - Goal : 사용자는 여러 개의 재무 목표를 설정할 수 있는 **1:N 관계**이다.

User - Type : 사용자는 하나의 **Type**에 속하며, 유형은 여러 사용자를 가질 수 있는 **N:1 관계**이다.

- 성능 고려
 - **Foreign Key** 기반 관계 설계를 통해 데이터 무결성을 유지하고, 사용자 삭제 시 연관 데이터 삭제를 용이하게 하였다.
 - 자주 조회되는 컬럼[user_id, spend_date]을 기준으로 인덱스를 적용하여 사용자별 소비 내역 및 기간별 조회 성능을 개선하였다.
 - 소비 데이터는 집계 및 분석 빈도가 높아, 날짜 컬럼을 기준으로 범위 조회가 가능하도록 **DATETIME** 타입을 사용하였다.
 - **Notification** 및 **Goal** 테이블은 사용자 단위 조회가 많아 **user_id** 기준 조회 최적화를 고려하여 설계하였다.

4.4 API 설계

Path	Methods	View	결과
/spending-analysis/	GET	spending_analysis	spending_analysis.html
/spending-history/	GET	spending_history_view	spending_history.html
/spending-type/	GET	spending_type	spending_type.html
/assets/	GET	asset_detail	asset_detail.html
/mypage/	GET	mypage	mypage.html
/profile/edit/	GET	profile_edit	profile_edit.html
/goal/set/	POST	set_goal	목표 저장 후 redirect
/mydata/start/	GET	mydata_start	mydata_start.html
/login/	GET/POST	login_view	로그인 폼/처리
/signup/	GET/POST	signup_view	회원가입 폼/처리
/logout/	GET	logout_view	로그아웃 후 redirect

5. 기능 설계 및 구현 상세

5.1 기능 1: 가상 마이데이터 생성

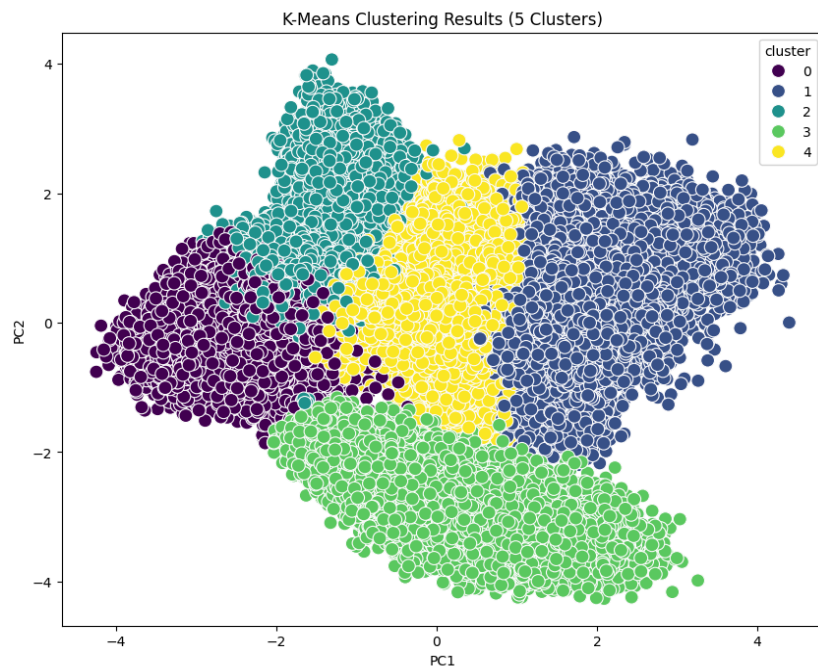
- 목적: 실제 금융 마이데이터 활용이 제한된 환경에서 여러 사용자의 소비 내역을 분석하고 소비 유형을 진단하기 위한 기초 데이터를 확보하고자 한다. 다양한 소비 성향을 반영한 사용자 페르소나 기반의 소비 데이터를 생성함으로써, 소비 분석, 소비 유형 분류, 이상소비 탐지 등의 주요 기능을 검증·구현할 수 있는 데이터 환경을 구축하는 것을 목적으로 한다.
- Input: userID
- 처리 로직
 - 여러가지 페르소나에 맞는 소비 내역 데이터를 생성, 데이터는 **spend_date**, **method**, **price**, **details**, **transaction_type** 의 이름을 가진 컬럼으로 구성
 - 생성된 데이터의 **details** 컬럼을 JEMINI API에 입력하여 각각의 **category** 컬럼 분류
- Output : DB - spending table
- 스크린샷/결과

spending_id	user_id	category	spend_date	method	price	details	transaction_type	memo
1	167	etc	2023-12-31 15:00:00	-	121693753	잔불이월	입금	
2	167	coffee	2024-01-01 02:38:00	카드	27900	테라로사	출금	
3	167	shopping	2024-01-01 12:01:00	카드	18700	올리브영(생필품)	출금	
4	167	food	2024-01-02 01:56:00	카드	63800	마켓컬리	출금	
5	167	shopping	2024-01-03 04:55:00	카드	89500	에이블리	출금	
6	167	shopping	2024-01-03 08:53:00	카드	77600	한성	출금	
7	167	beauty	2024-01-03 09:56:00	카드	59500	에스테틱	출금	
8	167	coffee	2024-01-03 10:43:00	카드	13700	폴바셋	출금	
9	167	food	2024-01-03 12:07:00	카드	41900	이자카야	출금	
10	167	coffee	2024-01-04 04:01:00	카드	5600	감성개인카페	출금	
11	167	living	2024-01-04 06:17:00	카드	102700	관리비	출금	
12	167	transport	2024-01-04 07:26:00	카드	244700	타이어프로	출금	
13	167	transport	2024-01-04 07:32:00	카드	77000	HD현대오일뱅크	출금	
14	167	shopping	2024-01-04 12:28:00	카드	16800	올리브영(생필품)	출금	
15	167	shopping	2024-01-05 01:09:00	카드	88600	지그재그	출금	
16	167	health	2024-01-05 02:38:00	카드	42200	약국	출금	
17	167	transport	2024-01-05 11:37:00	카드	248000	타이어프로	출금	
18	167	health	2024-01-05 12:29:00	카드	206700	삼성화재다이렉트	출금	
19	167	food	2024-01-05 13:07:00	카드	2400	세븐일레븐	출금	
20	167	shopping	2024-01-06 01:00:00	자동이체	4900	데이터플러스멤...	출금	
21	167	entertai...	2024-01-06 04:33:00	카드	137100	클로연습장	출금	
22	167	food	2024-01-06 12:53:00	카드	36600	로기요	출금	
23	167	coffee	2024-01-06 13:37:00	카드	3300	이디야	출금	
24	167	food	2024-01-07 01:17:00	카드	18700	동네식자재마트	출금	
25	167	shopping	2024-01-07 02:18:00	카드	64000	지그재그	출금	
26	167	transport	2024-01-07 07:21:00	카드	10100	지역물택시	출금	
27	167	living	2024-01-07 08:40:00	카드	82200	도시가스	출금	
28	167	health	2024-01-07 15:15:00	카드	30000	정형외과	출금	

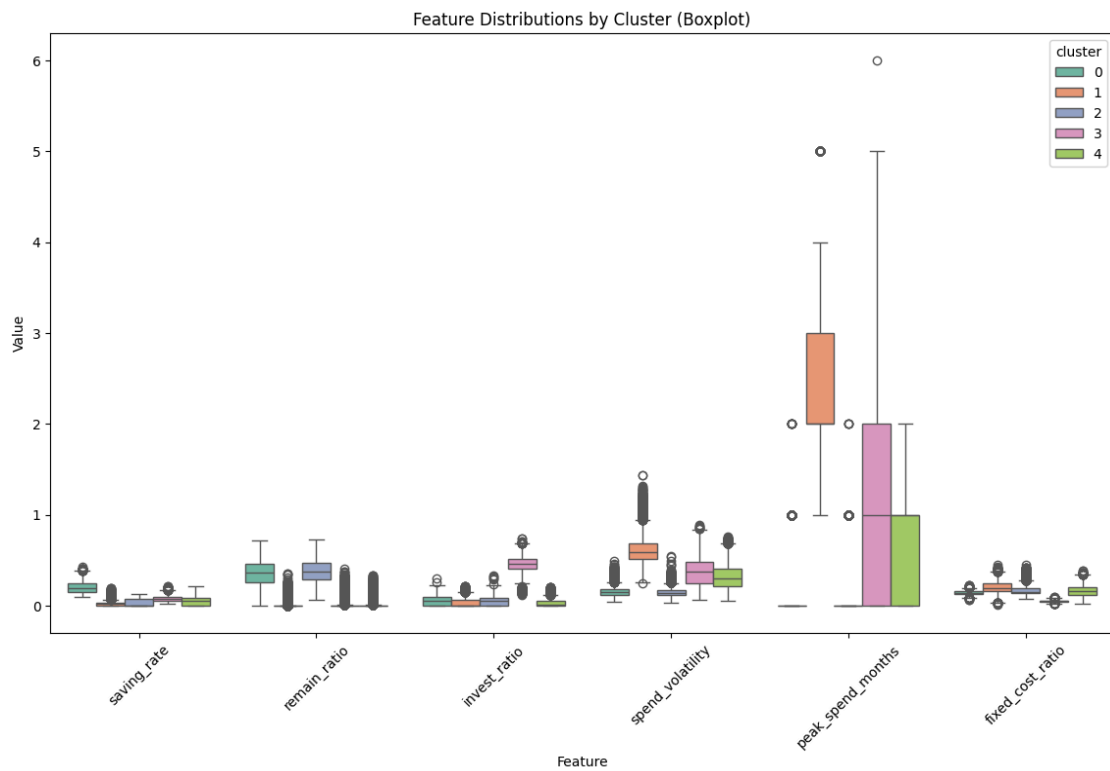
5.2 기능 2: 소비 유형 분류

- 목적: 사용자가 자신의 소비성향을 직관적으로 이해할 수 있도록 복잡한 소비 데이터를 단순화된 소비 유형으로 분류하여 제공하는 것을 목적으로 한다. 소비 유형 분류를 통해 사용자가 자신의 소비 성향을 쉽게 인지할 수 있도록하고, 사용자 기반 협업 필터링을 통해 사용자 유형에 따라 소비 성향에 맞는 금융상품을 추천한다.
- Input: model, userID, DB - Spending table
- 처리 로직
 - Moni userID에 해당하는 Spending 내역을 Feature Vector로 변환
 - 사전에 정의된 5가지 소비 유형(절약형, 안정형, 목표요정형, YOLO형, 진격의 투자형)에 대해 120,000개의 Feature Vector 군집화
 - 군집화된 Feature Vector를 학습한 Classification model Load
 - 학습된 모델로 Classification predict 수행
 - 회원가입 후 첫 로그인 시 survey를 통해 사용 경험이 있는 금융 상품을 수집
 - 유사한 사용자의 선택 데이터를 기반으로 협업 필터링을 적용해 유형별 선호도가 높은 금융 상품을 도출
 - YOLO형 사용자의 경우, 사용자 기반 협업 필터링을 적용하기보다는 소비 성향을 고려해 저축을 유도하는 전략으로 적금 상품을 추천하는 방식으로 구현
 - 사용자가 한눈에 확인할 수 있도록 버튼 클릭 시 해당 금융상품 페이지로 이동
- Output: DB - User table Spending Type, 금융상품 추천

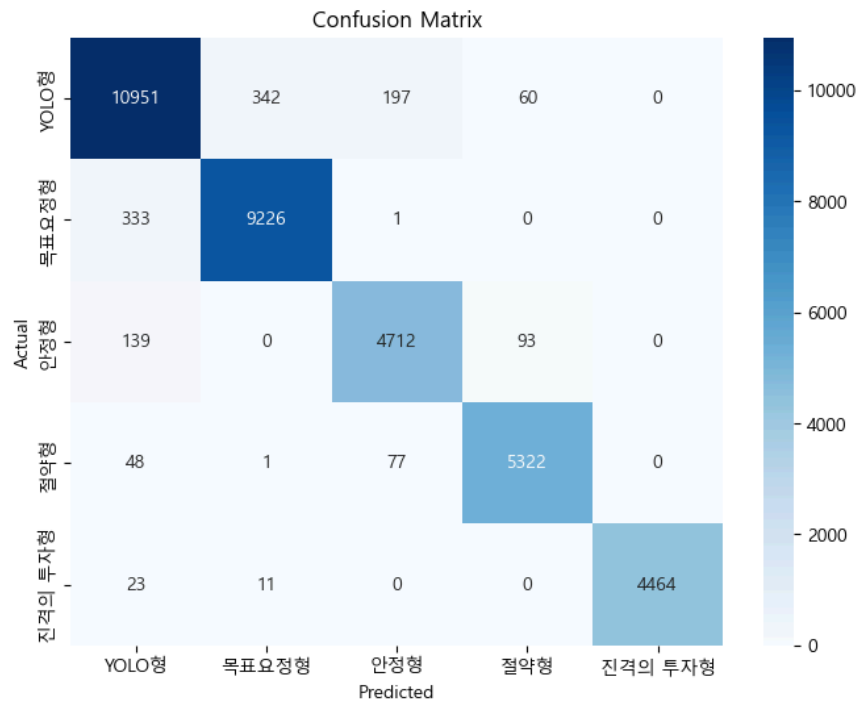
- 스크린샷/결과
 - 군집화된 Feature Vector



- 군집화된 클러스터별 특성 요약



- RandomForest 분류모델의 Confusion Matrix



5.3 기능 3: 이상소비 탐지

- 목적: 과거 소비 패턴을 기준으로 최근 소비의 이상 여부를 쉽게 판단하기 위해 설계되었다. 과거 전체 소비 데이터를 기반으로 정상적인 소비 범위를 정의하고, 최근 소비가 해당 범위를 벗어났는지를 통계적으로 판단함으로써, 사용자가 인지하지 못한 소비 이상 상황을 알림 형태로 제공하는 것을 목적으로 한다.
- Input: userID, DB - Spending table
- 처리 로직
 - Spending DB에서 userID에 해당하는 출금내역만을 가져와 하루 전까지의 데이터를 통해 사분위수 계산
 - 하루 전 데이터에서 Category당 상한경계 ($Q3 + 1.5IQR$) 이상의 데이터 확인
 - 이상치가 없는 경우, 이상치가 없다는 알림 생성
 - 이상치가 존재하는 경우, threshold값 이상/미만의 데이터에 따른 알림 생성
- Output: DB - Notification table
- 예외 처리: 해당 날짜에 이미 이상소비 알람이 있는 경우 똑같은 알람이 생성되지 않도록 예외처리
- 스크린샷/결과

어제 entertainment에서 93,362원만큼 과소비가 발생했어요 T.T
 어제 living에서 평소보다 소비가 높았어요
 어제 food에서 81,462원만큼 과소비가 발생했어요 T.T
 어제 food에서 평소보다 소비가 높았어요
 어제는 충동적인 소비가 없었어요 >.<!

	category	today_sum	upper_fence	is_anomaly
0	food	40000.0	23262.5	True
1	living	47000.0	172625.0	False
2	transport	3200.0	2975.0	True

5.4 기능 4: 머신러닝을 통한 소비 예측

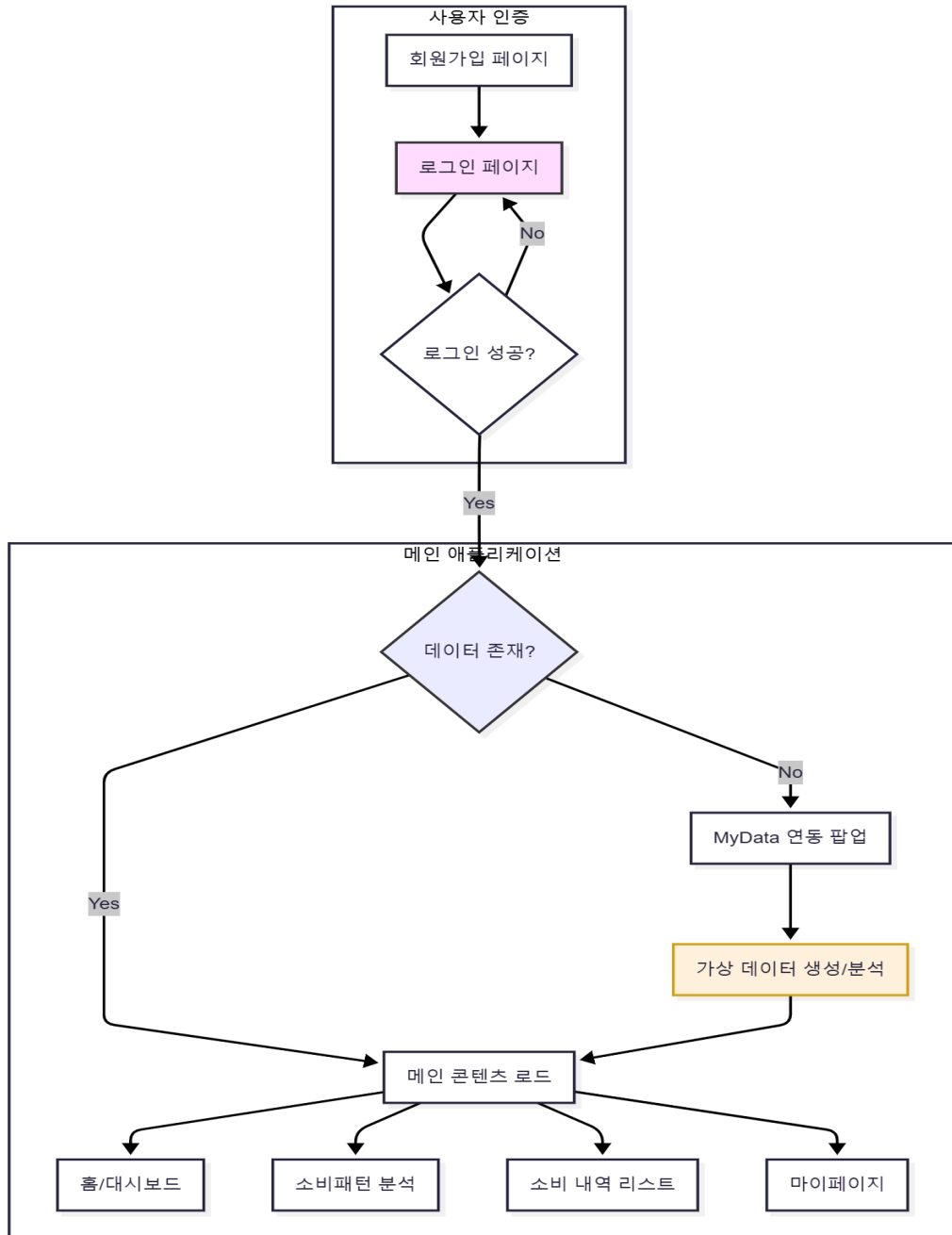
- 목적: 사용자의 과거 소비 행동 데이터를 분석하여 향후 소비 금액 및 소비 패턴을 예측함으로써, 미래의 소비를 사전에 인지하여 합리적인 소비 계획을 할 수 있도록 지원하는 것을 목적으로 한다. 이를 통해 과거 소비 분석을 넘어, 미래 소비에 대한 코칭 기능을 제공하고자 한다.
- Input: userID, DB - Spending table
- 처리 로직
 - 데이터 중 transaction_type이 출금에 해당하는 데이터만 filtering
 - 소비 날짜 데이터 기준 시간 관련 파생 변수(hour, weekday, is_weekend) 생성
 - category 및 price 데이터를 전처리하여 학습 가능한 형태로 변환
 - XGBoost 회귀 모델 & Lightgbm 모델 통한 다음 주 소비 금액 예측, 성능비교
 - 다음 주 예산 초과 발생 확률 예측
- Output: model prediction, DB - notification table
- 예외 처리: 데이터가 3개월 미만일 경우 ML 모델 training 수행하지 않게 함
- 보안사항 : 소비 확률은 정확도를 구했지만, 금액은 데이터 부족으로 오차값 큼
- 스크린샷/결과

	week_start	week	month	category	actual_spend	expected_spend	pred_spend_flag	pct_diff
0	2024-12-16	51	12	health	91800	441672.000000	1	-381.124183
1	2024-12-16	51	12	shopping	386690	322744.031250	1	16.536753
2	2024-12-16	51	12	food	40800	162568.937500	1	-298.453278
3	2024-12-16	51	12	transport	244400	95872.867188	1	60.772149
4	2024-12-16	51	12	entertainment	31300	85549.609375	1	-173.321436
5	2024-12-16	51	12	coffee	24500	20313.802734	1	17.086519
6	2024-12-16	51	12	savemoney	0	7626.853027	0	NaN
7	2024-12-16	51	12	etc	0	1065.825684	0	NaN
8	2024-12-16	51	12	investment	0	1003.593018	0	NaN
9	2024-12-16	51	12	beauty	0	942.790649	0	NaN
10	2024-12-16	51	12	living	0	764.396423	0	NaN

6. 구현 결과

6.1 화면(UI) 구현 결과

- 화면흐름도



- 실제 구현 화면

[회원가입]

[로그인]

Moni

Moni

로그인

아이디 찾기 | 비밀번호 찾기

처음 오셨나요?
회원가입

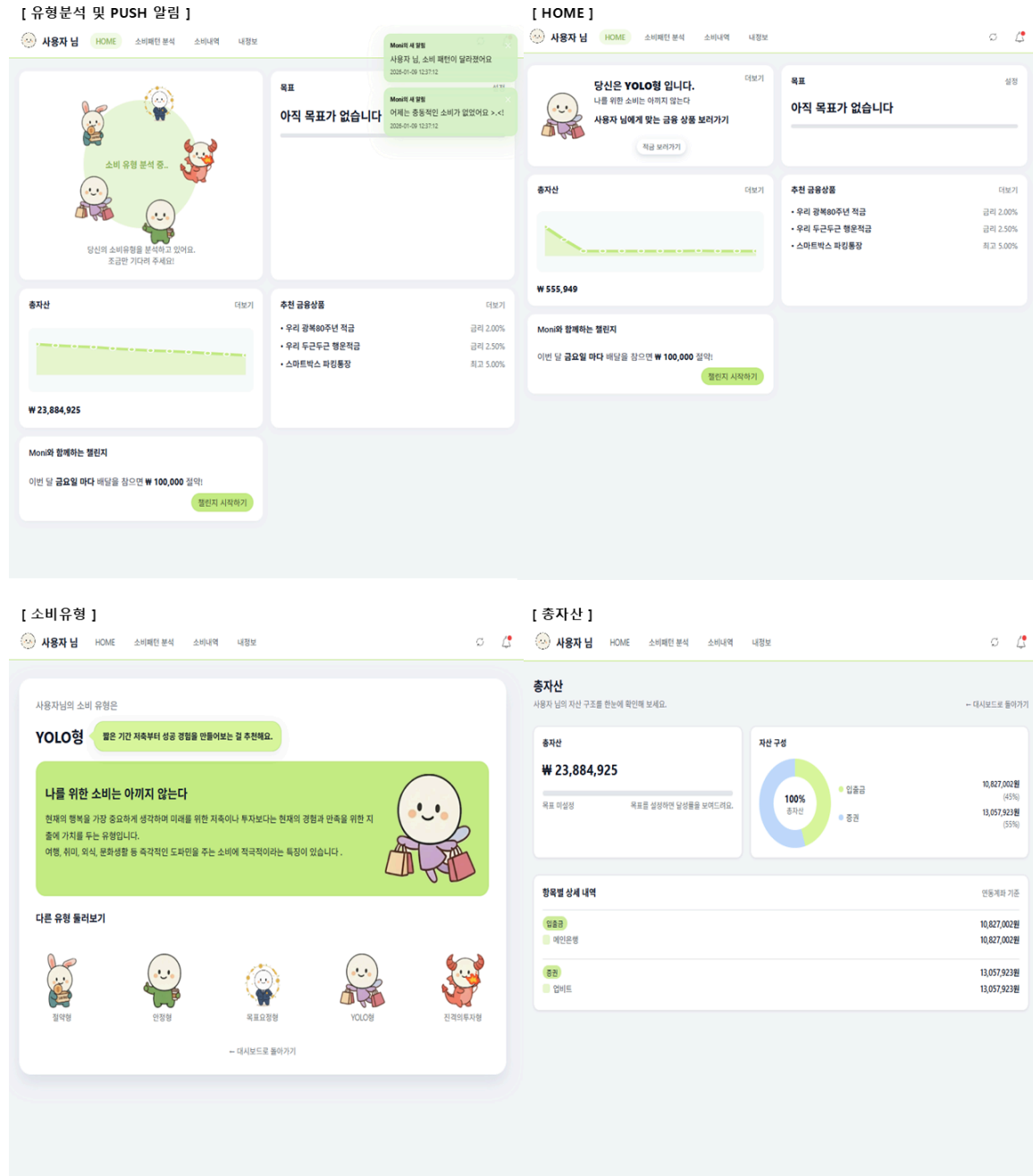
N 네이버 로그인 카카오 로그인



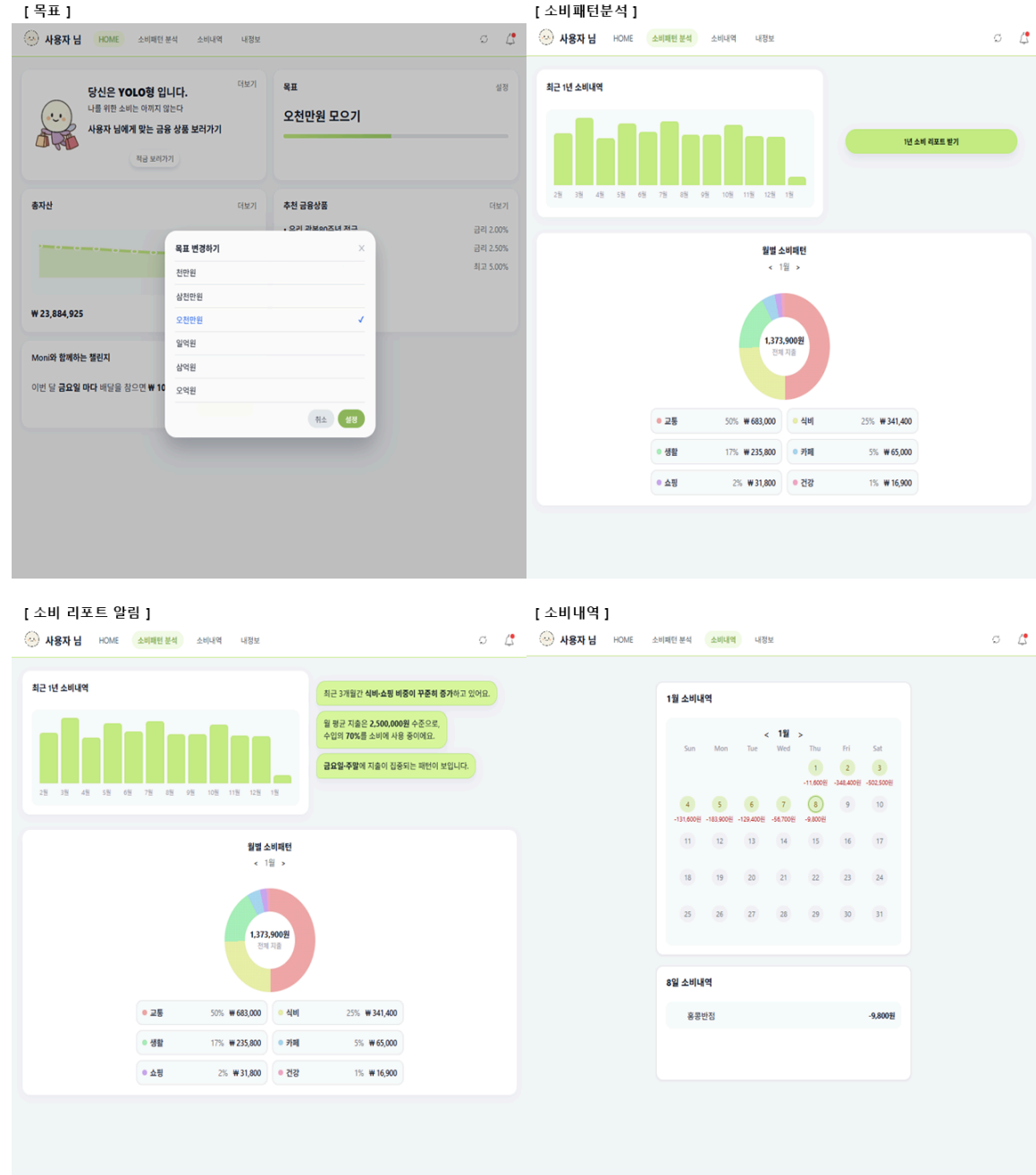
회원가입 완료

【 설문조사 】

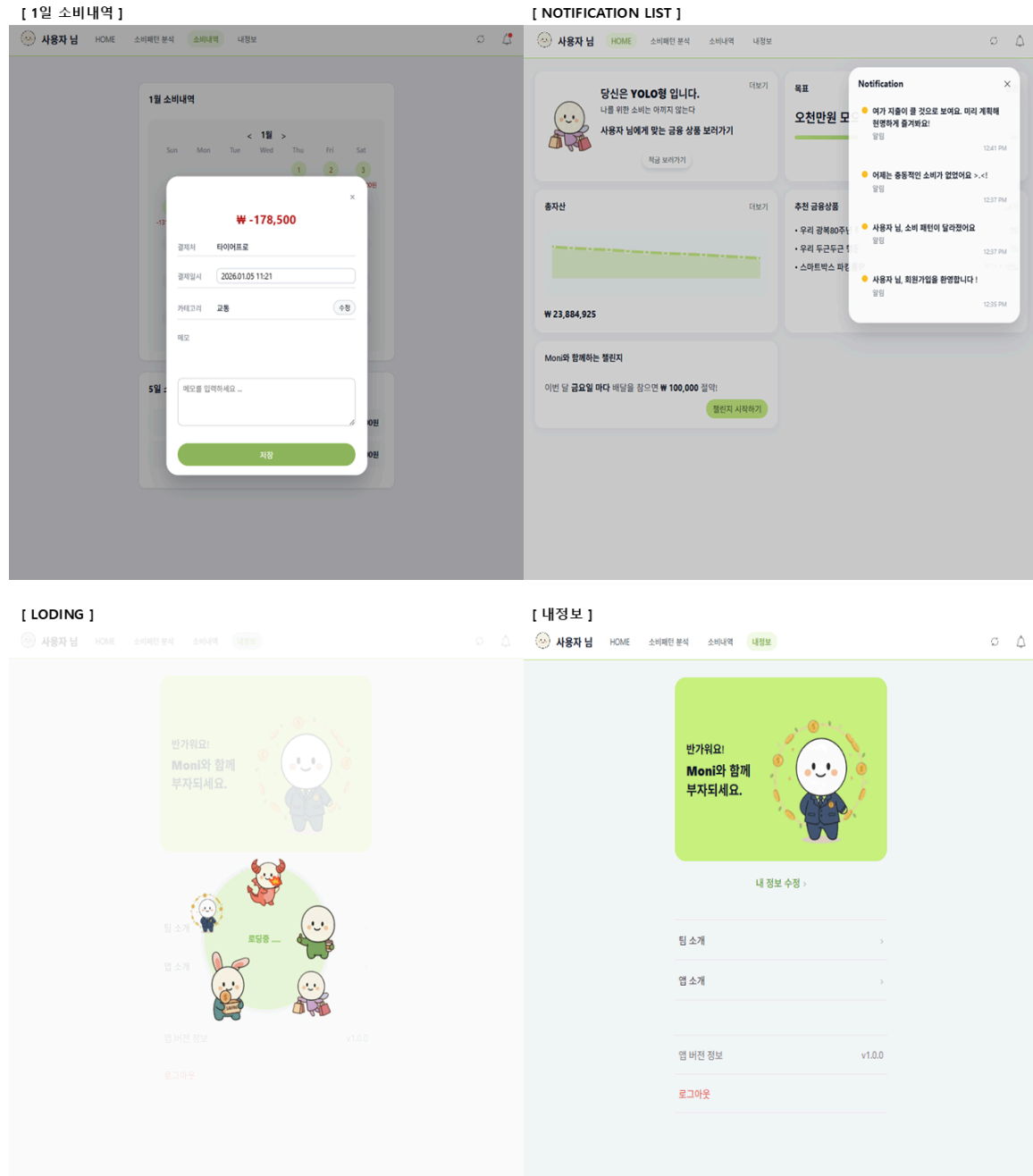
[가상데이터 생성]



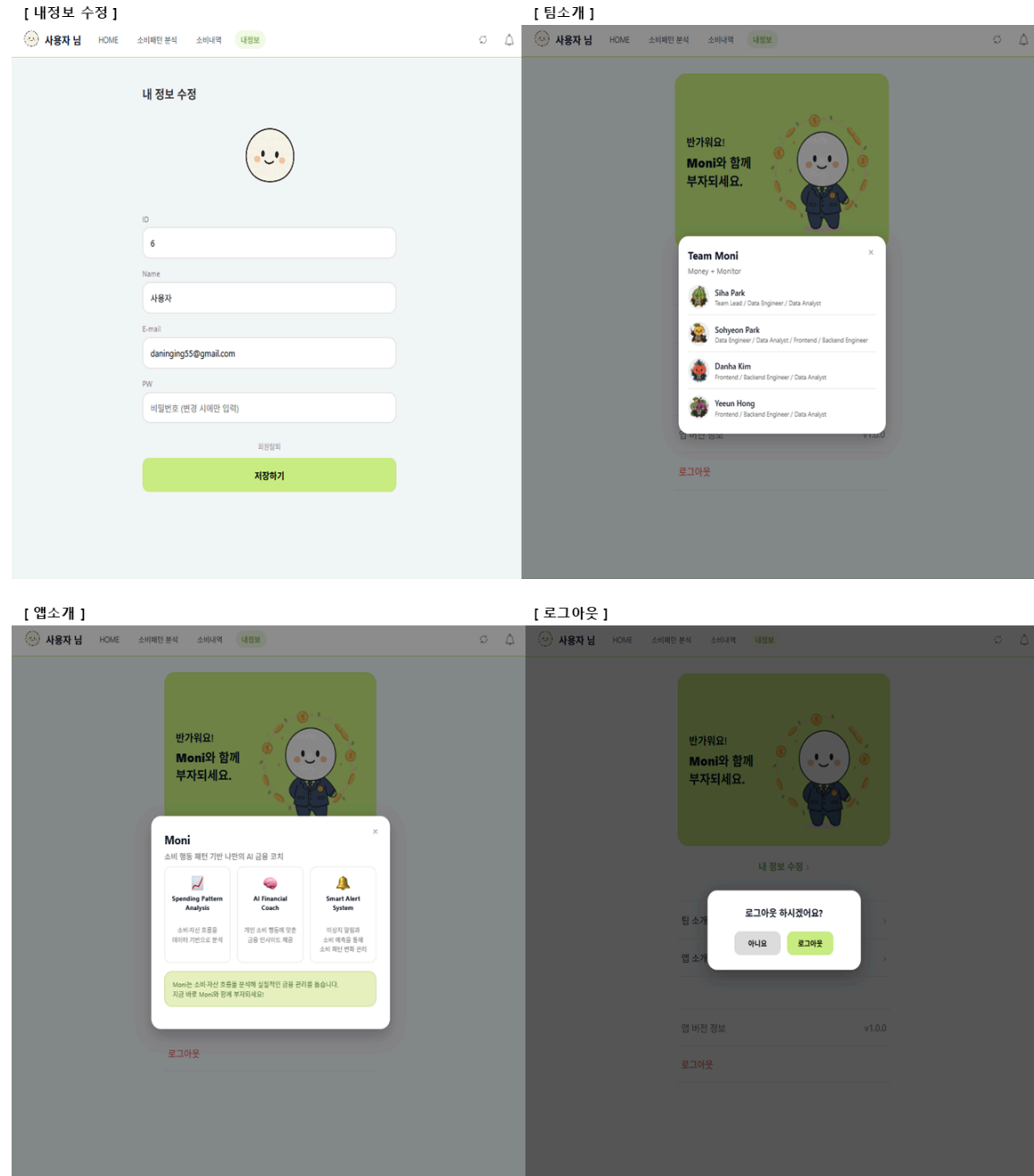
Moni, Personal AI Financial Coach Based on Spending Behavior Patterns



Moni, Personal AI Financial Coach Based on Spending Behavior Patterns



Moni, Personal AI Financial Coach Based on Spending Behavior Patterns



6.2 배포/운영

- 배포 환경

서비스 유형 : Web Application

Prod (운영): AWS EC2 (Ubuntu 22.04 LTS), RDS (PostgreSQL/MySQL).

Web Server: Gunicorn 및 Nginx를 통한 서비스 배포

형상 관리 : Git 기반 소스 코드 관리

배포 방식 : 도메인 기반 수동 배포

- 운영 모니터링

웹 서버 로그 기반 서비스 상태 모니터링

7. 개인별 작업 내용

7.1 박시하

- 담당 영역: Data Engineer, Data Analyst, Backend
 - 주요 기여
 - 데이터 파이프라인 구축
 - 가상의 사용자의 소비내역데이터 생성
 - LLM을 통한 소비내역 데이터 카테고리 분류 모델 구축
 - K-means 군집화를 활용한 사용자 소비유형 분류
 - RandomForest 분류 모델 학습
 - 머신러닝 결과 분석, 사용자 알림문구 생성용 LLM모델 구축
 - 미래 소비 예측&예방을 위한 알림 생성 로직 구축
 - AWS RDS 연동
 - AWS EC2를 통한 배포
 - Gunicorn, Nginx를 통한 배포
 - 산출물
 - 📁 가상데이터 예시
 - 📄 모델 정의서 & 성능평가서
 - 📄 함수정의서
 - 사용 기술
 - Cloud: [EC2, RDS / AWS]
 - Data Analyst: [Pandas, sklearn(Kmeans, RandomForestClassifier) / Visual Studio Code]
 - Database: [MySQL / MySQL Workbench]
 - 문제 해결
- [금융마이데이터 사용불가능]

- 이슈: 금융마이데이터 사용불가능이슈
- 원인: 금융사업자가 없는 팀프로젝트의 단위로는 금융마이데이터API 사용불가능
- 해결: 프로젝트 단위에서 금융마이데이터API 사용이 불가능하여 같은 형식의 가상데이터를 직접 생성(120,000 건)하여 K-means군집화 모델과 RandomForest 분류 모델을 학습


[모델 실행시간]

- 이슈: LLM모델을 통한 카테고리화 모델의 실행시간 20H 이슈
- 원인: 소비 내역을 일일이 JEMINI API를 통해 카테고리에 대한 질문
- 해결: 이미 질문한 내역을 하나의 사전처럼 파일로 저장해두는 구조로 변경, 모델의 실행시간 단축(20H -> 1m)

· 배운 점:

- 해당 프로젝트를 통해 서비스 기획, 구현, 배포까지 모든 과정을 경험할 수 있는 기회를 통해 각 부분에 관한 여러 지식을 얻을 수 있었고, 팀 프로젝트를 통해 다른 사람들과의 협업에 대한 경험도 쌓을 수 있었다. 또한 팀장으로 팀을 이끄는 관리자라는 직책에 대한 경험 까지 여러 가지 경험을 얻어 갈 수 있는 프로젝트였다.

7.2 김단하

- 담당 영역: Frontend, Backend, Data Analyst
- 주요 기여
 - 프로젝트 초기 환경 구축 및 구조설계
 - 홈, 소비유형, 총자산, 목표, 소비패턴분석, 소비내역, 분석중, loding, notification list, notification push, survey pop up화면 UI 구현
 - 소비패턴분석 결과 및 시각화 연동, 목표 총자산 패턴분석 소비내역 DB연결, notification list, notification push DB연결, 백엔드 API 설계, 프론트엔드 연동 구현
 - XGBoost 기반 소비 예측 모델 설계 및 학습
- 산출물
 - UI [6.1 화면구현결과와 실제 구현 화면 참조]
 - 함수정의서
 -  XGBoost
- 사용 기술
 - Frontend: [HTML, CSS, JavaScript, Django / Visual Studio Code]
 - Backend: [Python, Django, Requests / Visual Studio Code]
 - Data Analyst: [Pandas, NumPy, XGBoost / Visual Studio Code]
 - Database: [MySQL]

· 문제 해결



[XGBoost 기반 소비 예측 모델 한계 및 개선]

- 이슈: **XGBoost Regressor** 기반 소비 금액 예측에서 오차가 과도하게 커 예측 결과의 신뢰도 저하
- 원인: 소비 데이터 양이 충분하지 않고, 소비가 발생하지 않는 주(금액 = 0)가 다수 존재하는 데이터 구조로 인해 **MAE, MAPE** 지표가 의미 없는 수준으로 왜곡되어 소비 금액을 직접 예측할 경우 예측값이 0으로 수렴하거나 비정상적으로 큰 값으로 튀는 현상이 발생함
- 해결: 소비 금액을 직접 예측하는 단일 회귀 구조를 포기하고, 소비 발생 여부(0/1)를 예측하는 분류 모델(**Classifier**)과 소비가 발생한 경우에만 금액을 예측하는 회귀모델(**Regressor**)을 분리한 **Two-stage** 구조로 모델을 재설계했다. 이를 통해 서비스 알림에서는 금액 예측 대신 해당 카테고리에서 소비가 발생할 확률 기반 예측 결과를 제공하도록 개선

· 배운 점

- 본 프로젝트를 통해 기획 단계부터 개발 환경 구축, 데이터 분석, 서비스 연동, 그리고 배포에 이르기까지 전반적인 개발 흐름을 직접 경험하며 서비스 구조를 이해하고 구현할 수 있었다. 팀 프로젝트를 수행하며 각 역할을 맡은 팀원들과 소통하고 조율하며 방향을 맞춰가는 과정이 흥미로웠고, 협업이 분석 결과와 서비스 설계의 완성도를 높이는 데 기여한다고 느꼈다.

7.3 박소현

- 담당 영역 : 프론트/백엔드 엔지니어, 데이터 엔지니어, 데이터 분석
- 주요 기여
 - 소비유형, Notification, Home UX 개선을 위한 UI 구조 변경
 - 회원가입 기능 구현
 - 데이터 파이프라인 구축 및 MySQL Workbench를 사용한 DB 구축
 - Notification, User DB 백엔드 연동
 - Threshold를 기준으로 학습용 가상데이터 분류 알고리즘 구현
 - RandomForest Classifier 분류 모델 구축
 - 사용자의 가상데이터 생성 구현
 - 가상데이터 기반 소비내역 Feature Vector 추출 기능 구현
 - RandomForest Classifier 모델을 사용한 소비 유형 분류 구현
 - 매일 전일 데이터에 대한 이상치 감지 시 알림 생성 로직 구현
 - 소비 유형별 적합한 금융상품을 추천 및 추천 상품의 금융 서비스 페이지로 연동 기능 구현
- 산출물
 -  모델 정의서 & 성능평가서
 -  함수정의서
- 사용 기술
 - Frontend: [Django/HTML/Visual Studio Code]
 - Backend: [Django/Requests/Python/Visual Studio Code]
 - Data Analyst : [RandomForest Classifier]
 - Database: [MySQL/MySQL Workbench]

· 문제 해결

[회원가입 기능 이슈]

- 이슈 : 회원가입 이후 사용자 데이터가 정상적으로 연동되지 않는 문제 발생
- 원인 : 초기 설계 단계에서 커스텀 사용자모델 (MoniUser)을 기준으로 DB및 백엔드 개발 진행
이후 Django 기본 인증 모델(auth_user)이 마이그레이션 과정에서 생성되며 사용자 테이블 이원화
- 해결 : Django 기본 인증 흐름은 auth_user로 유지하면서 회원가입 시 생성되는 auth_user 정보를 기반으로 동일한 식별자를 갖는 MoniUser 레코드를 함께 생성하도록 로직 정리
이후 모든 도메인 데이터는 MoniUser를 기준으로 ForeignKey를 참조하도록 구조 통일

[Django ORM 날짜 필터링으로 인한 이상치 알림 중복 생성 이슈]



- 이슈 : 이상치가 감지될 경우 알림을 생성하는 로직에서 같은 날 동일 유형의 이상치 알림이 여러번 (로그인시마다) 생성되는 문제가 발생
요구사항 상 이상치 알림은 하루에 한 번만 제공되어야 하였음
- 원인 :

```
Notification.objects.filter(  
    user=moni_user,  
    notification_time__date=today_date  
)<code>).exists()
```

기존에는 위와 같은 Django ORM 방식으로 당일 알림 존재 여부를 체크하려고 시도하였으나, DATETIME Field + DB Timezone 처리, 로직에서의 날짜 경계, ORM의 __date 변환이 DB레벨에서 제대로 동작하지 않는 문제 등으로 인해,
이미 생성된 알림이 있음에도 exists()가 False로 판단되는 경우 발생

- 해결 : **ORM**의 추상화 계층에서 발생하는 불확실성을 제거하기 위해 **DB** 기준 날짜 비교과 명확한 **Raw SQL** 쿼리 방식으로 전환
DB 기준으로 정확한 판단이 가능해져 이상치 알림이 하루에 한번만 생성되는 로직을 안정적으로 보장함
- 배운점
 - 본 프로젝트에서는 비전공자 팀원들과 함께 협업하며 개발을 진행했다는 점에서 이전과는 다른 배움을 얻을 수 있었다. 개발 담당으로서 분석 로직 구현과 기능 연동을 맡아 서비스 구조를 설계하고 구현에 기여하였으며, 특히 기술적 배경이 다른 팀원들과의 협업 과정에서, 분석 결과를 서비스 흐름에 자연스럽게 반영하기 위한 소통과 조율의 중요성을 체감할 수 있었다.

7.4 홍예은

- 담당 영역: Frontend / Backend / Data Analyst
- 주요 기여
 - 회원가입, 로그인/로그아웃, 소비내역, 내정보 관리/수정, 서비스 소개 화면 UI 구현
 - 회원가입, 로그인/로그아웃, 소비내역 기능 구현, Frontend UI 및 서버 연동
 - 일·주 단위 소비 데이터를 기반 IQR 방식의 이상치 탐지 로직 구현 및 이상 소비 알림 기준 정의
 - LightGBM 기반 시간대별 소비 예측 모델 설계 및 학습
- 산출물
 -  LightGBM.ipynb
 -  Outlier.ipynb
 - UI [6.1 화면구현결과와의 실제 구현 화면 참조]
- 사용 기술
 - Frontend: Django/HTML/CSS/Visual Studio Code
 - Backend: Python/Django/Requests/Visual Studio Code
 - Data Analyst: Python/LightGBM

- 문제 해결

[LightGBM 기반 소비 예측 모델 적용 및 성능 안정화 문제]

- 이슈 : 특정 시간대에 예측 결과가 과도하게 집중되는 현상 발생
- 원인 : 원시 소비 데이터가 실제 소비가 발생한 시점만을 포함하는 구조로 구성되어 있어, 소비가 발생하지 않은 시간대의 패턴을 모델이 학습하지 못함. 또한 일반적인 **Random split** 방식의 데이터 분할은 시간 순서를 고려하지 않아 미래 예측 문제에 적합하지 않은 한계 존재
- 해결 : 전체 기간을 1시간 단위(**Time Slot**)로 재구성하여 소비가 발생하지 않은 시간대까지 학습 데이터에 포함함. 소비 발생 여부를 **Binary Target(spend: 0/1)**로 정의하여 문제 구조를 단순화함. **Logistic Regression**을 **baseline** 모델로 설정하고 **LightGBM**과의 성능 비교를 통해 모델 개선 효과를 검증함으로써, 개인의 시간대별 소비 패턴을 보다 안정적으로 학습하는 예측 모델 구조를 확립하고 서비스 알림 적용 가능성을 확보함.

- 배운 점

- 본 프로젝트를 통해 **Frontend, Backend, Data Analysis** 전 과정을 직접 경험하며 데이터 분석 결과가 서비스 기능과 **UI**로 효과적으로 연결되기 위해서는 데이터 처리 구조, 분석 단위 설계, 결과 전달 방식이 체계적으로 설계되어야 한다는 것을 배웠다. 또한 이상치 탐지와 소비 예측 모델링을 수행하며 분석 단위와 기준 설정이 모델 성능뿐 아니라 실제 서비스 활용도에 직접적인 영향을 미친다는 점을 학습하였고, 이를 통해 데이터 분석은 독립적인 결과 도출이 아니라 사용자 경험 개선과 의사결정을 지원하는 과정임을 인식하게 되었다.

8. 결론 및 회고

8.1 문제점 분석

금융 마이데이터 API 사용이 제한된 환경에서 가상 소비 데이터를 생성하여, 실제 환경에서 발생하는 변수의 다양성에 한계가 있었다. 약 12만 건의 데이터를 활용했음에도 유형 분류 모델의 학습 성능이 높게 나타났으며, 이 점이 실제 사용자 환경에서의 신뢰도와 실효성에는 한계가 있을 것으로 판단된다. 또한 제한된 프로젝트 일정으로 인해 기획과 진행 과정에서 세부적인 완성도를 충분히 다듬기보다는 전체적인 기능 구현과 결과물 도출에 집중할 수밖에 없다는 구조적 제약이 있었다.

8.2 향후 계획

- 처리속도 개선
 - 기존 소비 데이터 분석 및 예측 기능의 전체적인 처리 속도를 개선 할 예정이다. 데이터 전처리 및 집계 과정의 병렬처리를 통해 분석 처리 시간을 단축하고, 예측 모델 적용 과정에서 연산 최적화를 수행함으로써 사용자 요청에 대한 응답 속도 개선을 목표로 한다.
- 추천시스템 고도화
 - 현재까지 구현된 금융상품 추천을 기반으로, 소비 유형 분석 결과를 반영한 금융상품 추천 기능을 사용자의 소비형태에 맞는 구체화된 기능으로 확장하고자 한다. 이를 통해 소비 분석 결과가 단순한 정보 제공에 그치지 않고, 사용자의 금융 의사결정으로 자연스럽게 연결될 수 있는 서비스 구조를 완성하는 것을 목표로 한다.
- 금융마이데이터
 - 금융 마이데이터를 활용하여 데이터 범위 및 분석 한계를 개선할 계획이다. 현재는 가상 소비 데이터 기반으로 분석을 수행하고 있으나, 향후 마이데이터 연계를 통해 보다 다양한 데이터를 활용함으로써 소비 패턴 분석의 정밀도를 높이며 개인화된 소비 코칭 서비스를 고도화할 예정이다.

8.3 기대 효과

- 사용자 관점
 - 일, 주 단위의 소비 분석 및 예측을 통해 과거와 미래의 소비에 대한 개인화된 코칭을 제공받을 수 있다. 또한 금융 마이데이터를 기반으로 소비 내역 조회, 가계부 기록, 저축 목표 관리 기능을 사용함으로써 사용자가 자신의 소비 습관을 체계적으로 관리할 수 있다.
- 기술적 가치
 - 소비 내역을 단순 집계가 아닌 시계열 기반 행동 패턴 데이터로 변환하여 사용자별 소비 특성을 정량화하였다. 개인의 과거 소비 이력을 기준으로 한 모델링을 통해 상대적 이상 소비 및 변화를 탐지할 수 있다. 이와 함께 소비 발생 흐름을 반영해 매일 즉각적인 피드백을 제공한다. 또한, 분석 → 피드백 → 행동 변화 추적의 반복 구조를 통해 기존 서비스에서 부족했던 소비 패턴 교정을 지원한다.

9. 참고문헌

- 논문

머신러닝 기반 시계열 예측 시스템 비교 및 최적 예측 시스템 구현

- https://www.kci.go.kr/kciportal/ci/sereArticleSearch/ciSereArtiView.kci?sereArticleSearchBean.artid=ART003112113&utm_source=chatgpt.com

RandomForest

- <https://www.kci.go.kr/kciportal/ci/sereArticleSearch/ciSereArtiView.kci?sereArticleSearchBean.artid=ART002480029>

XGBoost, LightGBM을 이용한 종목추천 뉴스기사 기반 주식 수익률 방향성 예측

- <https://www.kci.go.kr/kciportal/ci/sereArticleSearch/ciSereArtiView.kci?sereArticleSearchBean.artid=ART003055332>

- 기술 블로그

GBM 그리고 XGBoost

- <https://www.youtube.com/watch?v=rVOYicsNgVQ>

WebView에서 웹 앱 빌드

- <https://developer.android.com/develop/ui/views/layout/webapps/webview?hl=k>

안드로이드 구조와 동작 원리

- <https://jtm0609.tistory.com/103>