



SSAFY 관통프로젝트

# 당신을 위한 금융 Partner: 사용자 맞춤형 금융 상품 비교 & 추천 Application

팀장 : 이지호

팀원 : 정태운



# 팀 업무 분담

이지호

- 회원 커스터마이징을 위한 User Model 구현
- 금융 상품 데이터 저장 및 조회, 리뷰 Model / 기능 구현
- 환율 데이터 요청 및 전송 기능 구현
- 커뮤니티 기능을 위한 글, 댓글 Model 구현
- 금융 상품 추천 알고리즘 구현
- 관리자 권한에서 금융 상품 수정 및 메일 전송 기능 구현



정태윤

- 메인 Page 구현
- 계정 관리 및 프로필 Page 구현
- 금융 상품 전체 / 상세 조회 Page 구현
- API 이용하여 근처 은행 정보 출력 기능 구현
- 커뮤니티 Page 구현
- 상품 추천 Page 구현



# 주요 이용 고객 및 서비스 목표

## 고객층

- 20대 후반 ~ 30대 초중반 청년
- 사회 초년생으로서 다양한 금융 정보 필요.
- 금융 정보의 복잡성과 바쁜 시간으로 인해 금융 정보를 쉽게 찾지 못함
- 다양한 금융 정보를 한 눈에 볼 수 있는 플랫폼이 필요

## 서비스 목표

- 예금 / 적금 금리 비교 및 상품 가입 기능
- 환율 계산 기능
- 주변 은행 검색 기능
- 고객 맞춤 금융 상품 추천 기능
- 금융 정보 공유를 위한 커뮤니티 기능

# 목차



Component 구조도



데이터베이스 모델링(ERD)



회원 커스터마이징,  
프로필 페이지



예/적금 금리 비교



환율 계산 기능



커뮤니티 기능



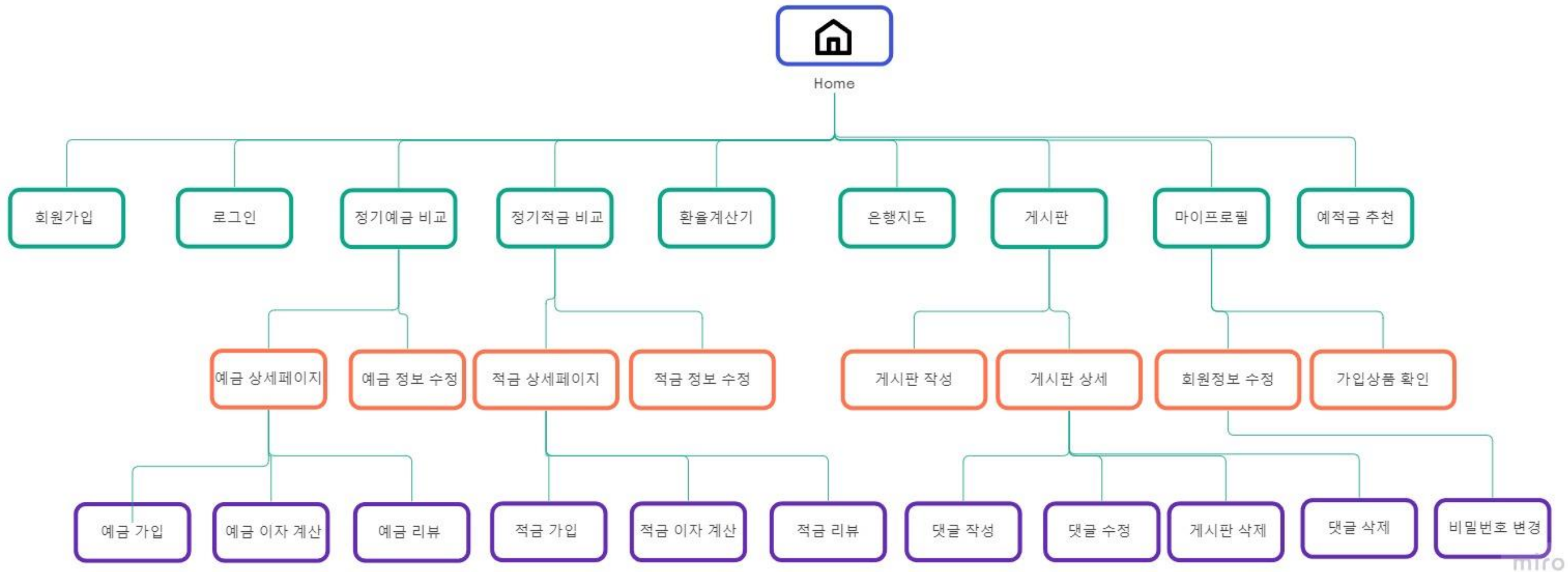
금융 상품 추천 알고리즘



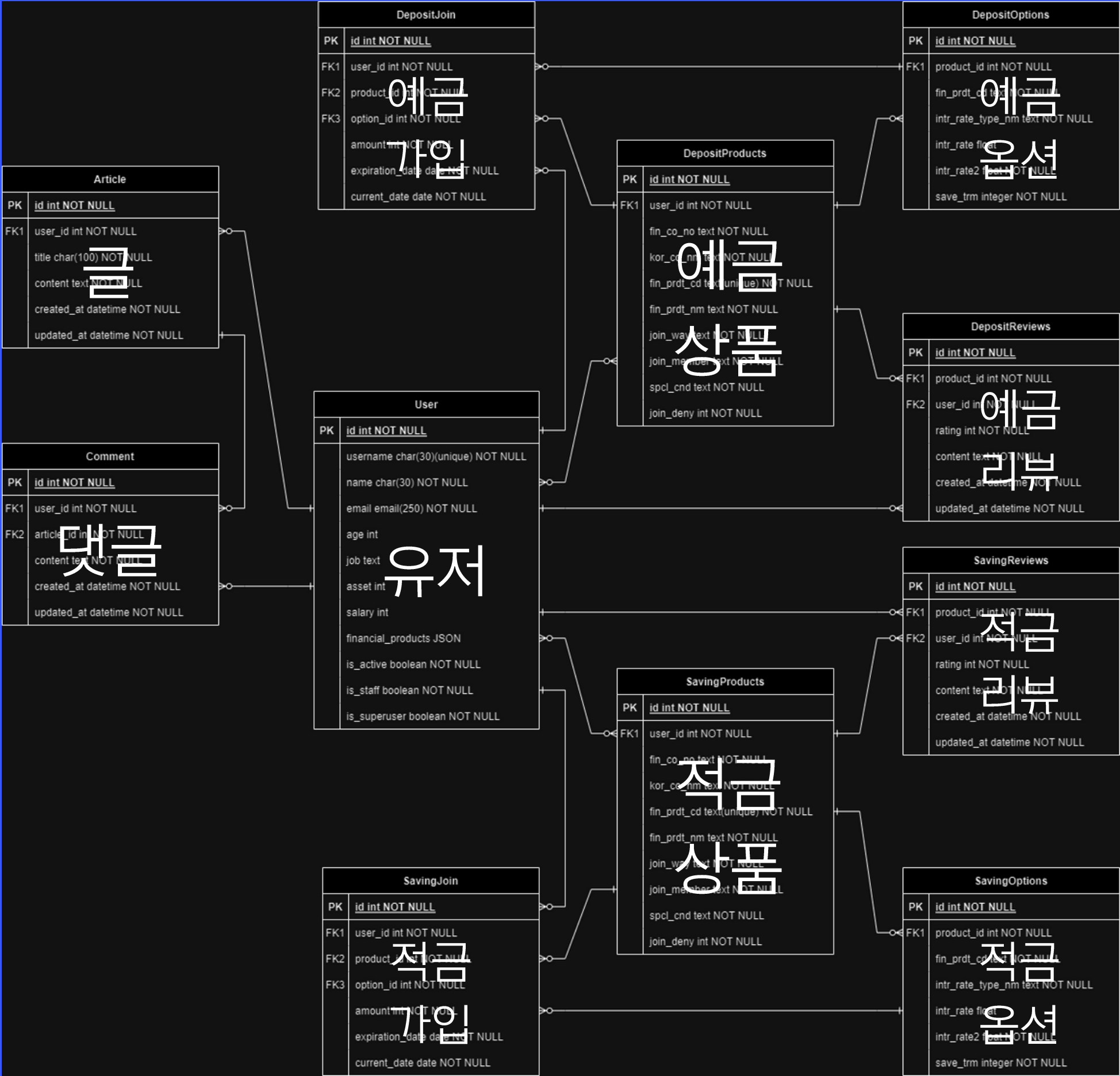
시연 / Q&A



# 1 Component 구조도







```
class User(AbstractUser):
    username = models.CharField(max_length=30, unique=True) # 유저 ID
    name = models.CharField(max_length=30) # 이름
    email = models.EmailField(max_length=250) # Email
    age = models.IntegerField(blank=True, null=True) # 나이
    job = models.TextField(blank=True, null=True) # 직업
    asset = models.IntegerField(blank=True, null=True) # 자산
    salary = models.IntegerField(blank=True, null=True) # 연봉
    financial_products = models.JSONField(blank=True, null=True) # 가입한 상품 목록
```

4

## 예/적금 금리 비교 - 관리자 기능

DGB주거래우대예금(첫만남고객형) 3 3.6

3.16

4

3.95

2.75

3.75

3.75

3.68

3.58

3.48

3.6

3.05

3.1

3.78

JB 다이렉트예금통장 (만기일시지급식) 3.9 4.1

### 금리 정보 변경

6개월 (단리)

기본 금리

3.62

우대 금리

4

12개월 (단리)

기본 금리

3.58

금리 정보 변경

## ☆ 금리 수정 확인 - 상품명: 스마트모아Dream정기예금

^ 보낸사람 jeh01129@naver.com VIP

받는사람 wjdxodbs52@naver.com

2023년 11월 24일 (금) 오전 11:35

스마트모아Dream정기예금 상품의 금리가 다음과 같이 변경되었습니다!

가입 기간: 6 - 금리: 3.62 / 우대금리: 4.0

가입 기간: 12 - 금리: 3.58 / 우대금리: 3.78

가입 기간: 24 - 금리: 3.62 / 우대금리: 3.82

가입 기간: 36 - 금리: 3.68 / 우대금리: 3.88





## Exchanges

GET /banking/exchanges/

HTTP 201 Created

Allow: OPTIONS, GET

Content-Type: application/json

Vary: Accept

```
[
  {
    "result": 1,
    "cur_unit": "AED",
    "ttb": "349.25",
    "tts": "356.3",
    "deal_bas_r": "352.78",
    "bkpr": "352",
    "yy_efee_r": "0",
    "ten_dd_efee_r": "0",
    "kftc_bkpr": "352",
    "kftc_deal_bas_r": "352.78",
    "cur_nm": "아랍에미리트 디르함"
  },
  {
    "result": 1,
    "cur_unit": "AUD",
    "ttb": "839.23",
    "tts": "856.18",
    "deal_bas_r": "847.71",
    "bkpr": "847",
    "yy_efee_r": "0",
    "ten_dd_efee_r": "0",
    "kftc_bkpr": "847",
    "kftc_deal_bas_r": "847.71",
    "cur_nm": "호주 달러"
  },
]
```



## 6 커뮤니티 기능

### 커뮤니티

전체 3 건

글쓰기

No	제목	작성자	작성일
3	db저축은행 햇살론 너무 좋았네요!! 2	admin	2023. 11. 24.
2	[잡담]주거래 은행 어디쓰세요? 3	wjdxodbs52	2023. 11. 24.
1	은행에서 서명하다 지쳤어요. 3	admin	2023. 11. 24.

#### db저축은행 햇살론 너무 좋았네요!!



admin

2023-11-24 11:51:09

예니팀장님에게 상담 받으면서 적당한 대출 알아봤고 저는 db저축은행 햇살론을 사용해보게 되었어요^^ 결과 너무 만족하고 있는 중이고요~

댓글 2 >

댓글을 남겨보세요



admin

db저축은행 햇살론 승인 성공 축하드립니다! 정부지원상품이라 금리도 저렴하고 안정적이니 마음에 쏙 드셨을것 같아요! 어려울것 같던 대출이 이렇게 수월하게 진행되서 너무 다행이네요!

2023-11-24 11:51:18

admin

db저축은행 햇살론으로 대출 진행이 잘 되셔서 너무 다행이네요~ 기대 출 비율이 높아지다보니 추가한도 발생이 잘 안되어서 상담 요청 하셨나 봐요~ 최대한 그래도 회원님의 조건에 맞게 진행을 도움 받아 너무 다행 이십니다~

2023-11-24 11:51:25



# Categorical Variable Based Filtering

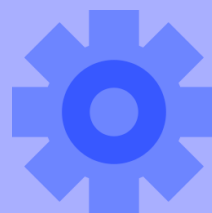
범주형 변수 기반 필터링



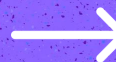
사용자의 개인적 특성과 유사한 그룹이 선호하는 아이템을 추천하는 알고리즘



사용자의 나이, 직업, 연봉, 자산을 기준으로 사용자 그룹 필터링



필터링된 각 사용자 그룹에서 인기 많은 상품 순으로 정렬하여 추천



# Item-Based Collaborative Filtering

## 다른 고객이 함께 구매한 상품



햇반 매일 찰잡곡밥, 210g, 24개

**25,370원** 로켓배송  
(100g당 503원)

★★★★★ (14,123)



햇반, 205g, 36개

**32,590원** 로켓배송  
(100g당 442원)

★★★★★ (13,317)



곰곰 광천 도시락김, 32개

**8,150원** 로켓배송  
(10g당 509원)

★★★★★ (63,208)



햇반 즉석밥 작은공기, 130g, 36개

**27,900원** 로켓배송  
(100g당 596원)

★★★★★ (22,594)



동원 고추 참치, 85g, 12개

**16,320원** 로켓배송  
(10g당 160원)

★★★★★ (40,561)



햇반 발아현미밥 작은공 130g, 36개

**34,880원** 로켓배송  
(100g당 745원)

★★★★★ (28,511)

# Item-Based Collaborative Filtering

아이템 기반 협업 필터링



넷플릭스, 아마존, 쿠팡에서  
사용 중인 알고리즘



사용자의 행동 패턴(아이템  
에 대한 평점)을 분석



사용자가 아직 평가하지 않은  
아이템의 선호도를 예측하여  
사용자가 선호할 만한 아이템을  
찾아내는 알고리즘



아이템 간의 유사도를 계산하여,  
사용자가 선호하는 아이템과  
유사도가 높은 아이템을 추천

# Item-Based Collaborative Filtering

- Cosine Similarity

$$\text{sim}(u, u') = \cos(\theta) = \frac{R_u \cdot R_{u'}}{\|R_u\| \|R_{u'}\|} = \frac{\sum_{i=1}^n R_{ui} \times R_{u'i}}{\sqrt{\sum_{i=1}^n (R_{ui})^2} \times \sqrt{\sum_{i=1}^n (R_{u'i})^2}}$$

- 사용자  $u$ 와 사용자  $u'$ 가 같은 방향성을 보고 있는지를 평가함
- 일반적으로 사용되는 유사도

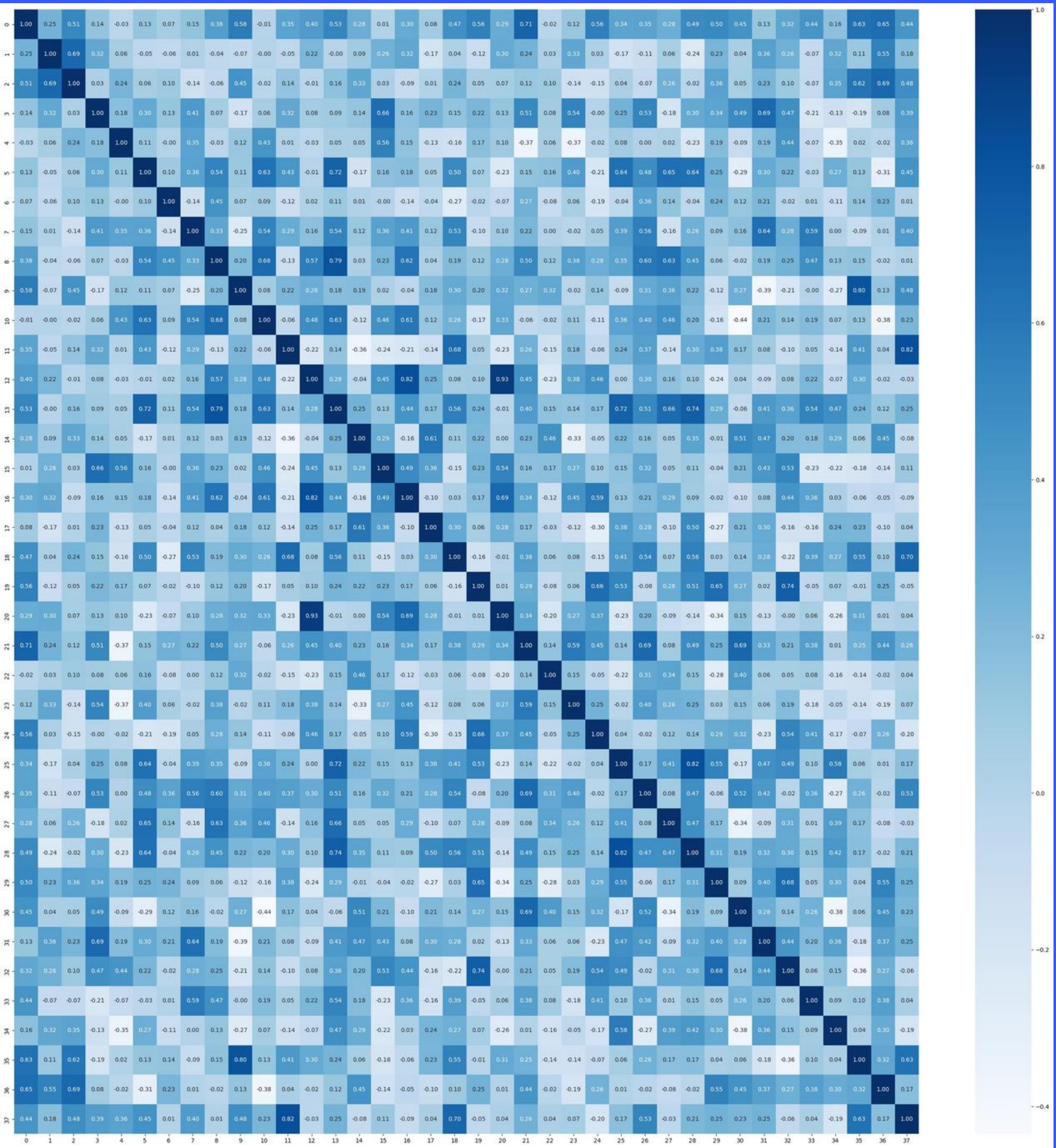




```
@api_view(['GET'])
@permission_classes([IsAdminUser])
def deposit_rating_matrix(request):
    # 사용자와 상품에 대한 평점 행렬을 생성
    reviews = DepositReviews.objects.all()
    df = read_frame(reviews, fieldnames=['user__id', 'product__id', 'rating'])
    df = df.rename(columns={'user__id': 'user_id', 'product__id': 'product_id'})
    df = df.groupby(['user_id', 'product_id']).mean().reset_index()
    rating_matrix = df.pivot(
        index='user_id',
        columns='product_id',
        values='rating'
    )
    # 상품 간의 Cosine 유사도 계산
    # 코사인 유사도는 두 벡터 간의 코사인 값을 사용하여 측정한 유사도로, 값이 1에 가까울수록 유사도가 높다는 것을 의미
    item_similarity = cosine_similarity(rating_matrix.T.fillna(0))
    deposit_item_similarity_df = pd.DataFrame(
        item_similarity,
        index = rating_matrix.columns,
        columns = rating_matrix.columns
    )
    return Response(deposit_item_similarity_df.to_dict(), status=status.HTTP_201_CREATED)
```



# 금융 상품 추천 알고리즘



# Cold-Start Problem

- 새로 가입한 유저의 경우, 해당 유저에 대한 정보를 수집할 수 없어 적절한 제품을 추천하지 못하는 문제
- 사용자가 평가한 상품이 없거나 아이템 간 유사도가 없는 경우, 가장 인기가 많은 아이템을 대신 추천하여 문제점을 해결함

```
# 사용자가 평가한 상품이 없다면 가장 인기가 많은 아이템을 추천함.
if not valid_product_ids:
    popular_items = item_similarity_df.sum().sort_values(ascending=False)
    recommended_items = popular_items.index[:item_numbers]
    return Response({'recommended_items': recommended_items.tolist()}, status=status.HTTP_201_CREATED)
# 사용자가 평가한 상품이 있다면, 해당 상품에 대한 유사도를 평균내어 사용자의 평점을 계산
user_ratings = item_similarity_df.loc[user_product_ids].mean().dropna()
sorted_user_ratings = user_ratings.sort_values(ascending=False)
# 사용자가 평가한 모든 아이템의 유사도가 0인 경우에는 가장 인기 있는 아이템을 추천
if user_ratings.sum() == 0:
    popular_items = item_similarity_df.sum().sort_values(ascending=False)
    recommended_items = popular_items.index[:item_numbers]
# 가장 높은 평점을 받은 상품과 유사한 상품을 찾아 추천
else:
    item_similarity_df = item_similarity_df.drop(user_product_ids, errors='ignore')
    top_item_id = sorted_user_ratings.index[0]
    similar_items = item_similarity_df[top_item_id].sort_values(ascending=False)
    recommended_items = similar_items.head(item_numbers).index
```







## 금융 상품 추천 알고리즘

기본 정보

상품 추천 받기 (선호도)

상품 추천 받기 (유저 정보)

### 상품 추천 받기

정태윤님만을 위해 준비한 특별한 추천  
당신의 선호도를 반영한 최적의 금융 상품을 찾아봤어요.

(정기예금)제주은행 - 제주Dream 정기예금 (개인/만기 지급식)

(정기예금)부산은행 - LIVE정기예금

(정기예금)우리은행 - WON플러스예금

(정기예금)주식회사 케이뱅크 - 코드K 정기예금

(정기예금)신한은행 - 쉬운 정기예금

(정기적금)경남은행 - 행복Dream적금

(정기적금)농협은행주식회사 - NH내가Green초특세상적금

(정기적금)토스뱅크 주식회사 - 토스뱅크 자유 적금

(정기적금)국민은행 - KB 특★한 적금

(정기적금)수협은행 - Sh월복리자유적금



# 프로젝트 소감

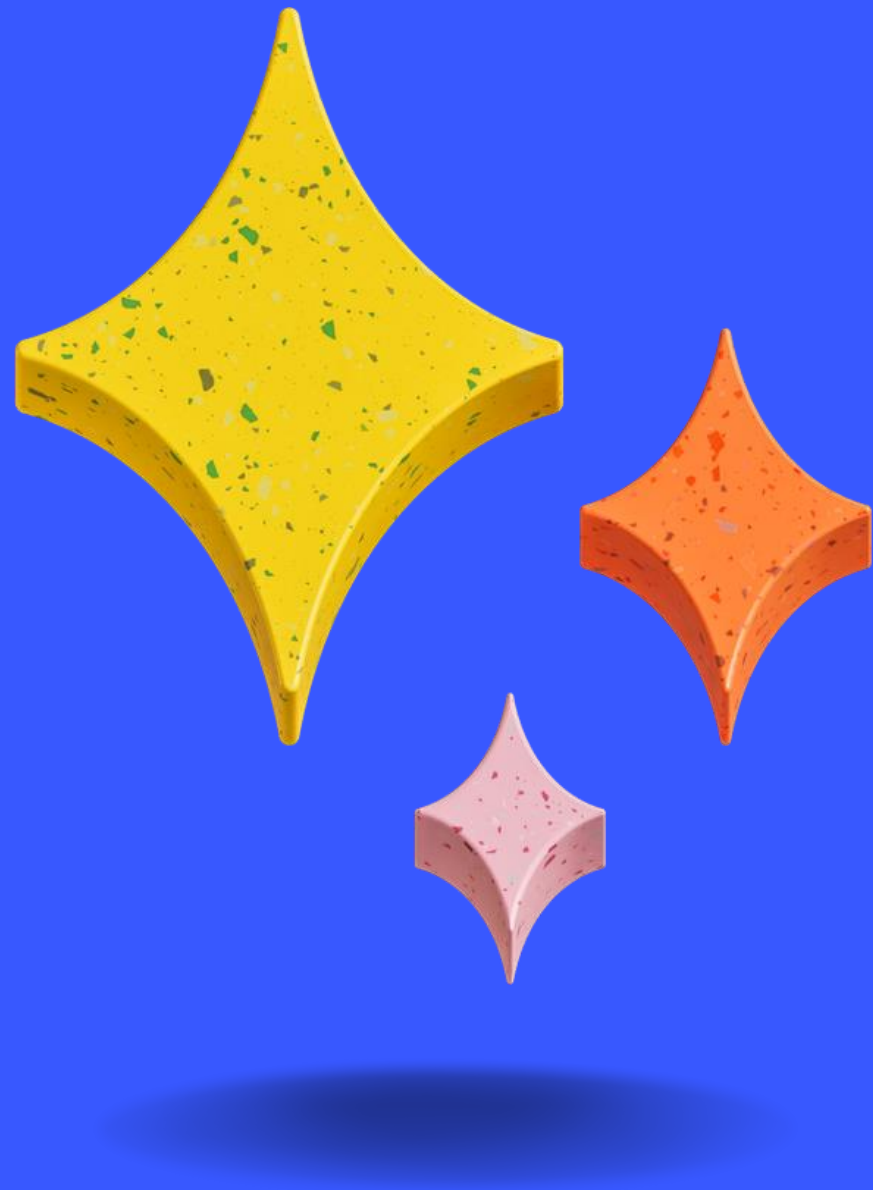
이지호

- Django REST Framework를 이용한 웹 개발, 추천 알고리즘 등 지금까지 배운 지식을 이용해 실제 서비스를 구현
- 사용자의 특성에 따른 상품 추천 알고리즘을 구현함으로써 개인화된 서비스 제공
- 프로젝트를 통해 문제를 해결하는 과정에서 새로운 도전과 문제들을 마주하게 되었고, 이를 해결하기 위해 끊임없이 배우고 성장하는 경험을 할 수 있었습니다.

정태윤

- Vue.js를 통해 데이터를 사용자에게 효과적으로 전달함으로써, 사용자 친화적이고 직관적인 인터페이스를 제공
- Vue.js의 컴포넌트 기반 구조를 이용해 페이지를 모듈화하고, 이를 통해 코드의 재사용성을 높이고 유지 관리를 편리하게 함.
- 또한, 동적 바인딩 기능을 통해 사용자의 상호작용에 따라 웹 페이지의 내용이 실시간으로 업데이트되는 점을 통해 사용자 경험을 크게 향상시킬 수 있었습니다.





# 감사합니다

## 시연 / Q&A