

재정조종단

투자 성향 기반 금융 상품 추천

서울 2반 이반이조
이규석
송창용

재정조종단

01 프로젝트 설명

도입배경
프로젝트 기능

02 팀 구성 및 역할

팀 구성, 역할 상세

03 기능 설명

데이터베이스
Django 구조

04 기능 상세

시연 예시

05 추천 알고리즘

머신러닝이란?
LightGBM

06 느낀점

소개할 내용을 입력해 보세요.
포함되는 내용을 나열해 보세요.

도입배경

도입 배경

코로나19 이후 비대면 거래 증가에 따라 **디지털 전환이 가속화**

디지털 자산관리가 **전 연령층으로 확대**
but 서비스는 비교적 단순한 수준이며 만족률이 높지 않음

고객 맞춤형 정보와 서비스를 선별하는 **개인화된 자산관리가 필요**



결과

비슷한 투자 성향, 조건을 가진 사람들을
기반으로 **투자 상품을 추천**

프로젝트 기능

투자 성향 기반 금융 상품 추천

투자 성향

얼마나 위험을 감수할 것인가?

재정 조건, 관심사,
개인정보....

사용자의 개인정보



금융 상품 추천

01 머신러닝을 활용한 종목 추천

LGBMClassifier

02 비슷한 조건을 가진 고객들의 상품 추천

Data Analysis

역할 분담

이규석

예금, 적금 상품
DataBase
Front-End
CRUD

금융 상품 API, DB 작성,
회원가입, 로그인, 마이페이지,
CSS, Vue

송창용

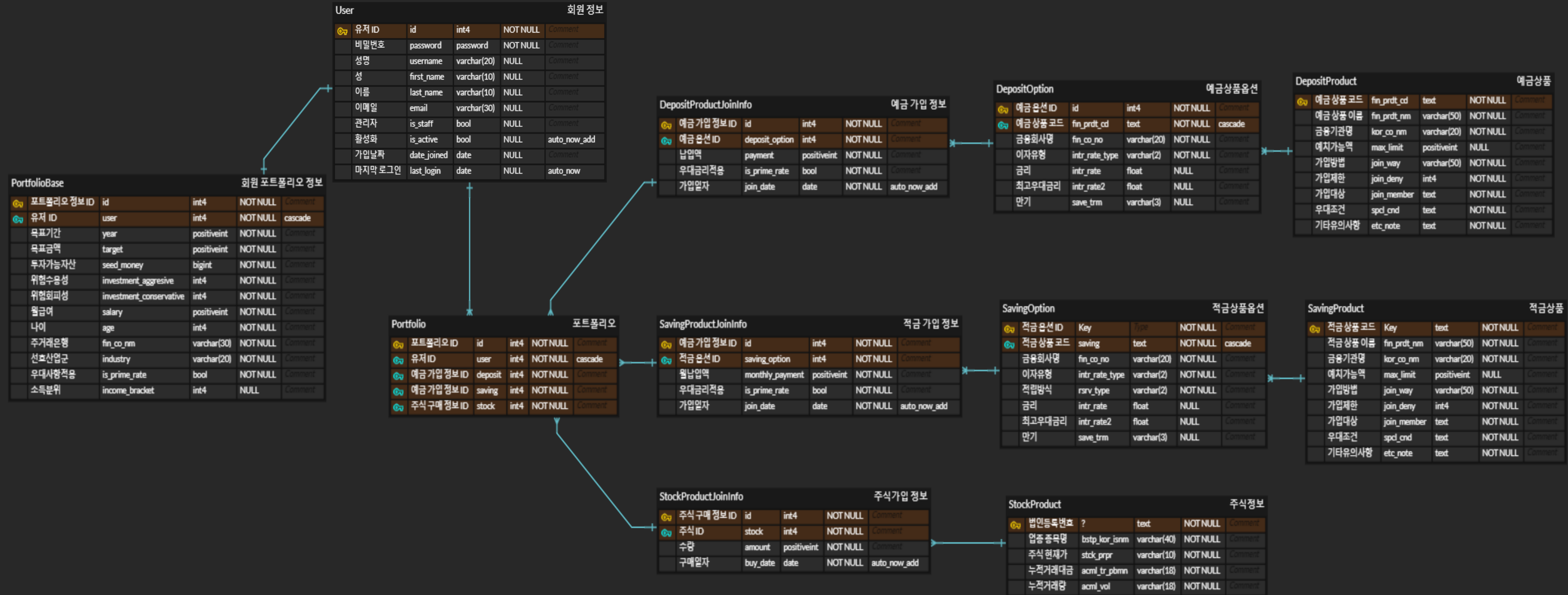
주식 상품
DataBase
추천 알고리즘 작성

주식 정보, DB 작성,
추천 알고리즘 구현,
Vue

같은 역할

상세 기능

데이터베이스



데이터베이스 - Portfolio

```
class PortfolioBase(models.Model):
    user = models.OneToOneField(get_user_model(), on_delete=models.CASCADE)
    year = models.PositiveSmallIntegerField()
    target = models.PositiveIntegerField()
    seed_money = models.BigIntegerField()
    invest_aggressive = models.IntegerField()
    invest_conservative = models.IntegerField()
    salary = models.PositiveIntegerField()
    age = models.PositiveSmallIntegerField()
    kor_co_nm = models.CharField(max_length=30)
    industry = models.CharField(max_length=20)
    is_prime_rate = models.BooleanField()
    income_bracket = models.IntegerField(blank=True, null=True) # 소득분위

class Portfolio(models.Model):
    user = models.ForeignKey(get_user_model(), on_delete=models.CASCADE)
    deposit = models.ManyToManyField(DepositProductJoinInfo, related_name='deposit_comb')
    saving = models.ManyToManyField(SavingProductJoinInfo, related_name='saving_comb')
    stock = models.ManyToManyField(StockProductBuyInfo, related_name='stock_comb')
```

PortfolioBase : 유저에게 금융 상품을 추천하기 위해 입력받는 정보

Portfolio : 유저에게 추천된 종목에 대한 정보

데이터베이스 - 예금

```
# 예금 상품
class DepositProduct(models.Model):
    fin_prdt_cd = models.TextField(unique=True)
    fin_prdt_nm = models.CharField(max_length=50)
    kor_co_nm = models.CharField(max_length=20)
    max_limit = models.PositiveIntegerField(null=True)
    join_way = models.CharField(max_length=50)
    join_deny = models.IntegerField()
    join_member = models.TextField()
    spcl_cnd = models.TextField()
    etc_note = models.TextField()

# 예금 상품 옵션
class DepositOption(models.Model):
    deposit = models.ForeignKey(DepositProduct, on_delete=models.CASCADE)
    fin_co_no = models.CharField(max_length=20)
    intr_rate_type = models.CharField(max_length=2, null=True)
    intr_rate = models.FloatField(null=True)
    intr_rate2 = models.FloatField(null=True)
    save_trm = models.CharField(max_length=3, null=True)
```

DepositProduct : 예금 상품(상품명, 한도, 가입 방법...)

DepositOption : 상품 정보(금리, 우대금리, 우대 조건...)

데이터베이스 - 적금

```
# 적금 상품
class SavingProduct(models.Model):
    fin_prdt_cd = models.TextField(unique=True)
    fin_prdt_nm = models.CharField(max_length=50)
    kor_co_nm = models.CharField(max_length=20)
    max_limit = models.PositiveIntegerField(null=True)
    join_way = models.CharField(max_length=50)
    join_deny = models.IntegerField()
    join_member = models.TextField()
    spcl_cnd = models.TextField()
    etc_note = models.TextField()

# 적금 상품 옵션
class SavingOption(models.Model):
    saving = models.ForeignKey(SavingProduct, on_delete=models.CASCADE)
    fin_co_no = models.CharField(max_length=20)
    intr_rate_type = models.CharField(max_length=2, null=True)
    rsrv_type = models.CharField(max_length=2)
    intr_rate = models.FloatField(null=True)
    intr_rate2 = models.FloatField(null=True)
    save_trm = models.CharField(max_length=3, null=True)
```

SavingProduct : 적금 상품(상품명, 한도, 가입 방법...)

SavingOption : 상품 정보(금리, 우대금리, 우대 조건...)

데이터베이스 - 주식

```
# 주식 정보
class StockProduct(models.Model):
    prdt_cd = models.CharField(max_length=10) # 종목 코드
    prdt_name = models.CharField(max_length=60) # 종목명
    end_price = models.IntegerField() # 조회일 종가
    fluctuation_rate = models.FloatField() # 등락률
    trade_amount = models.IntegerField() # 거래량
    trade_price_amount = models.IntegerField() # 거래대금
    capitalization = models.IntegerField() # 시가총액
    shared_amount = models.IntegerField() # 상장주식수
    idx_bztp_mcls_cd_name = models.CharField(max_length=60, blank=True) # 업종명
    one_before_end_price = models.IntegerField(blank=True, null=True) # 1년전 종가
    one_before_end_rate = models.FloatField(blank=True, null=True)
    two_before_end_price = models.IntegerField(blank=True, null=True) # 2년전 종가
    two_before_end_rate = models.FloatField(blank=True, null=True)
    three_before_end_price = models.IntegerField(blank=True, null=True) # 3년전 종가
    three_before_end_rate = models.FloatField(blank=True, null=True)
```

StockProduct: 주식 상품에 대한 정보
(조회일 기준 종가, N년전 종가, 거래 정보)

Django - Finance

```
urlpatterns = [
    path('create/data/', views.create_data), # 데이터 생성
    path('create/data/dummy/', views.create_dummy_data), # 더미데이터 생성
    path('create/data/dummy2/', views.create_dummy_data2), # 통합 더미데이터
    path('deposit/', views.deposit_list), # 적금정보
    path('saving/', views.saving_list), # 예금정보
    path('stock/', views.stock_list), # 주식정보
    path('myportfolio/', views.my_portfolio),
    path('user/product/', views.get_user_product_list) # 유저 - 금융 정보
]
```

create/data : 금융 상품 DB 생성

create/data/dummy/ : 예금, 적금, 주식별 더미 데이터 생성

create/data/dummy2/ : 통합 더미 데이터 생성

deposit/ : 적금 데이터 반환

saving/ : 예금 데이터 반환

myportfolio/ : 포트폴리오 데이터 반환

user/product/ : 임의의 유저 - 상품 가입 데이터 생성

Django - User

```
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path('accounts/', include('dj_rest_auth.urls')), # 로그인, 로그아웃  
    path('accounts/signup/', include('dj_rest_auth.registration.urls')), # 회원가입  
    path('accounts/portfolio/', views.portfolio), # 포트폴리오 데이터 생성  
]
```

admin/ : 데이터 관리
accounts/ : 로그인, 로그아웃
accounts/signup/ : 회원가입
accounts/portfolio/ : 포트폴리오 생성

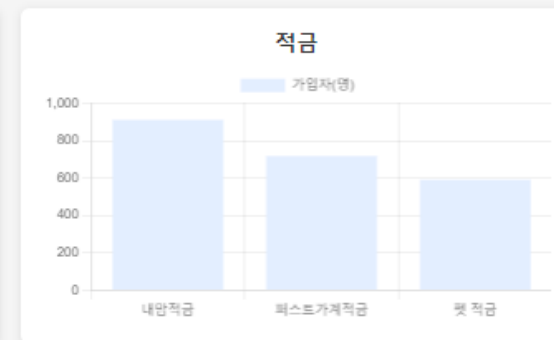
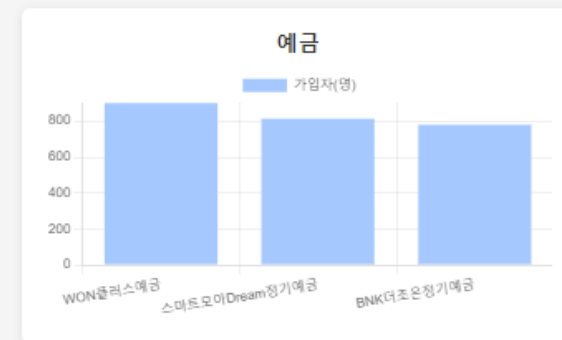
기능 상세 - 메인페이지

재정조종단(재종단)

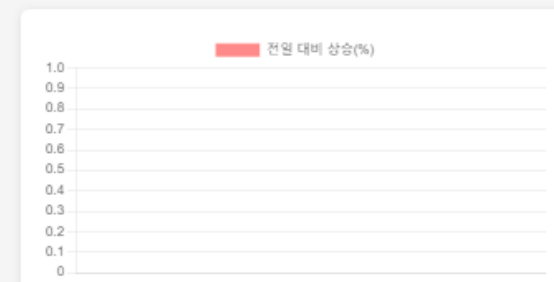
마이페이지 | 포트폴리오 추천받기 | 예금 | 적금 | 주식 | 로그아웃



가장 많은 사람들이 가입했어요



실시간 급등 종목



기능 상세 - 상품 추천

재정조종단(재종단)

마이페이지 | 비밀번호 변경 | 로그아웃

홈

주식

예금

적금

포트폴리오 추천 받기

인공지능이 나에게 맞는 상품을
추천합니다.

상품을 누르면 상세 페이지로 이동합니다.

예금: IM스마트예금

적금: 내맘적금

주식: 한미글로벌

나와 비슷한 사람은 아래 상품에
투자했어요!

예금 리스트

스마트모아Dream정기예금

DGB주거래우대예금(첫만남고객형)

IM스마트예금

적금 리스트

내맘적금

코드K 자유적금

우리SUPER주거래적금

주식 리스트

CJ씨푸드

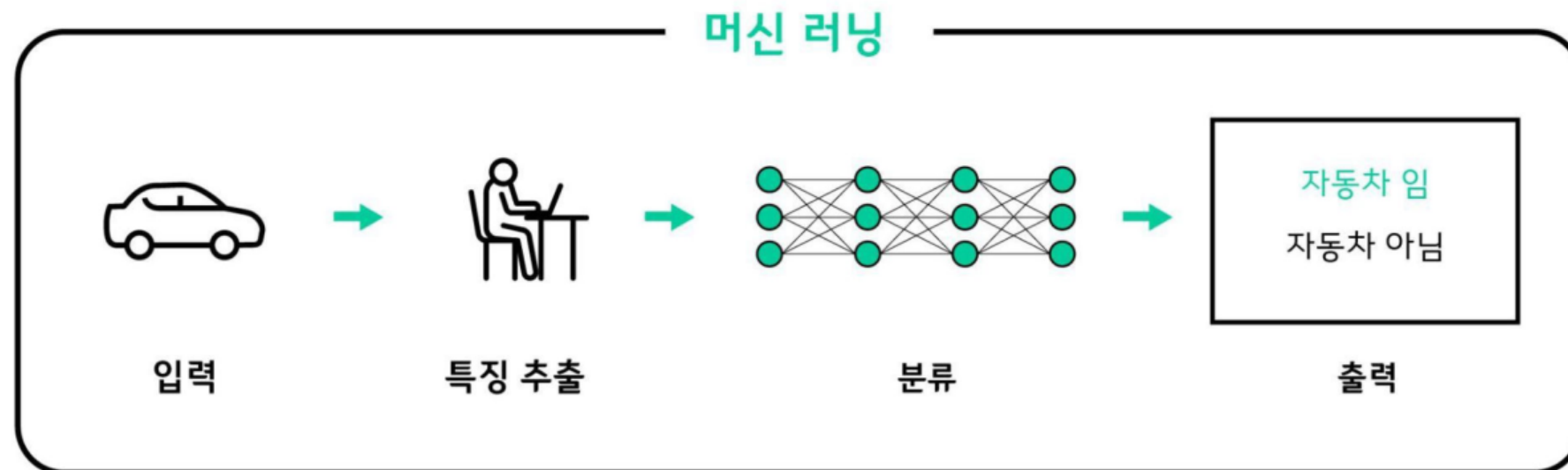
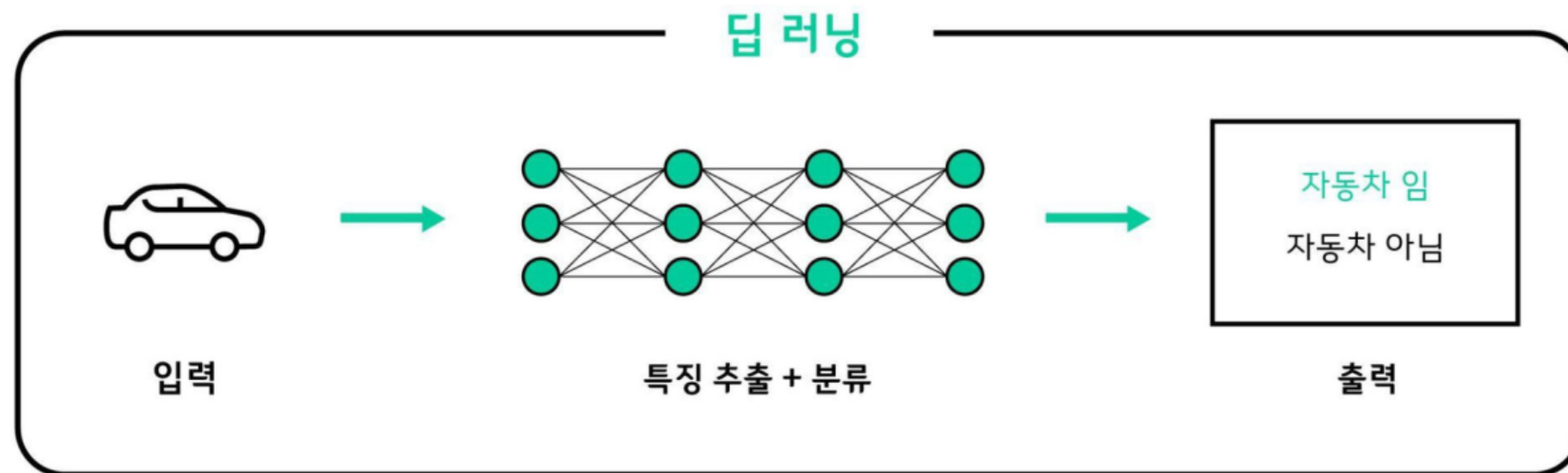
삼아알미늄

다우기술

추천 알고리즘

머신러닝이란?

딥 러닝과 머신 러닝 비교



머신러닝

컴퓨터가 명시적으로 프로그래밍 되지 않고도 데이터를 통해 특징을 학습하고 예측할 수 있는 인공지능의 한 분야

대량의 데이터를 분석하여 패턴을 발견하고,
이를 바탕으로 새로운 데이터에 대한 예측을 진행

사용배경

1. 딥러닝을 사용한 추천시스템은 사용자와 상품의 상호작용을 바탕으로 추천이 진행되는데 새로 가입한 유저는 이에 관한 정보가 없음
2. 사용자의 부가 정보(나이, 관심 산업, 재무 정보)를 학습에 활용할 수 있는 머신러닝이 효과적이라고 판단

추천 알고리즘

LightGBM



LGBM의 특징

트리 기반의 알고리즘
메모리를 적게 차지하고 속도가 빠름
결과의 정확성이 높음

느낀점

이곳에는 표의 내용을 요약하는 문장을 입력해 보세요.

항목	항목의 세부 내용
이규석	<p>지난 5개월동안 배운 내용을 모두 활용하여 하나의 완성된 프로젝트를 만들어서 의미있는 경험이었습니다.</p> <p>프로젝트를 진행하며 여러 문제들을 경험하고 직접 해결해나가면서 개발자로 한 단계 성장할 수 있는 시간이었습니다.</p>
송창용	<p>구현하고 싶은 기능이 많았는데 시간이 촉박해서 하고 싶었던 것을 모두 완벽하게 구현하지 못한 점이 아쉬웠습니다.</p> <p>프로젝트를 진행하면서 어떤 점이 부족하고 더 보완해야 할 지 알 수 있는 시간이었고, 2학기가 시작하기 전에 더욱 보완해나가야겠다고 생각했습니다.</p>