



랩업리포트 - 머글끼니

Wrap-Up Report, RecSys04 Team RECCAR

▼ 목차

Part 1. 프로젝트 Wrap Up

1-1. 프로젝트 개요

1-2. 프로젝트 팀 구성 및 역할

1-3. 프로젝트 수행 절차 소개

1-4. 자체 평가 의견

Part 2. 개인회고

김성연

배성재

양승훈

조수연

황선태

홍재형

▼ Part 1. 프로젝트 Wrap Up

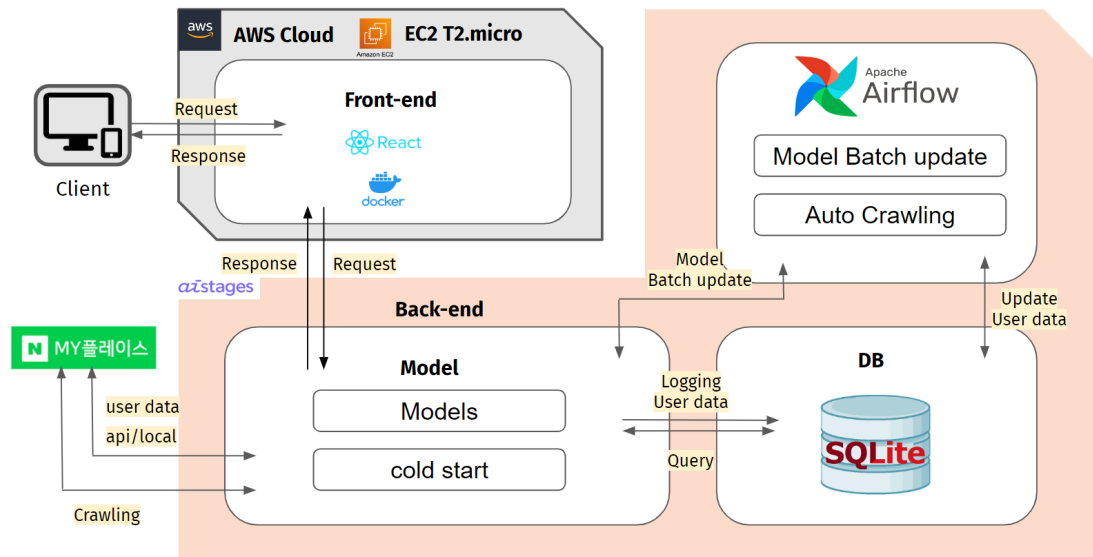
▼ 1-1. 프로젝트 개요

1-1-1. 프로젝트 목표

- 설명 가능한 추천
 - 어떤 이유에서 추천된 음식점인지 사용자에게 설명

- 개인화 추천
 - 회원가입 없이 네이버 플레이스 리뷰 내역만으로 개인 맞춤형 추천
- 모델 개선 자동화
 - Airflow를 통해 주기적인 크롤링과 모델 학습하는 환경 구축

1-1-2. 프로젝트 구조



1-1-3. 협업 도구

Slack

- 파일 공유, 논의해야할 사항을 Slack을 이용해 공유하였습니다.

Github

- Git Flow 브랜치 전략
 - master, 실험 branch를 나누어서 진행했습니다. 실험의 결과에 따라서 master branch에 merge하였습니다.
- Github Issues 기반 작업 진행
 - 팀원들이 진행하고 있는 실험의 세부 내용 공유를 위해 사용했습니다.
- Github Projects의 칸반 보드를 통한 일정 관리

팀원들이 현재 진행하고 있는 실험을 공유했습니다. 이를 통해 실시간 실험 관리에 유용했습니다.

Notion

실험결과 Parameter와 Recall@10을 쉽고 명확하게 공유하기 위해 노션을 활용했습니다.

1-1-4. 개발환경(AI Stage Sever)

- OS: Ubuntu 18.04.5 LTS
- GPU: Tesla V100-SXM2-32GB

▼ 1-2. 프로젝트 팀 구성 및 역할

- 김성연_T4040(팀장)
 - 모델링, 데이터베이스(SQLite) 데이터 전처리, metric 정의, Airflow
- 배성재_T4097(팀원)
 - 데이터 크롤링, 프론트엔드(React), 서비스 배포, 식당 좌표 수집, MLflow
- 양승훈_T4122(팀원)
 - 모델링, 백엔드(FastAPI), 서비스 배포, MLflow, Airflow
- 조수연_T4208(팀원)
 - 모델링, 백엔드(FastAPI), PPT
- 홍재형_T4233(팀원)
 - 데이터 크롤링, 데이터베이스(SQLite), 프론트엔드(React), 백엔드(FastAPI), Airflow
- 황선태_T4236(팀원)
 - 모델링, 프론트엔드(React), 발표

▼ 1-3. 프로젝트 수행 절차 소개

1-3-1. DATA

- 데이터 구조
 - 총 41460개의 식당, 382939명의 유저
 - 0.26퍼센트의 희소행렬 비율
- Data crawling
 - 네이버 플레이스에서 서울 모든 구의 각각의 동별로 인기있는 300개의 음식점의 방문 기록을 받아 옵니다.
 - 수집한 음식점에 사용자가 남긴 리뷰에 대한 정보를 수집합니다.
 - 동적 웹페이지를 크롤링하기 위해 음식점과 리뷰에 대한 정보는 셀레니움을 사용했습니다.
 - 식당의 x, y좌표는 네이버 검색 api가 있었고 셀레니움보다 속도가 빨라 api를 이용했습니다.
- Data preprocessing
 - random split : 유저의 리뷰 중 무작위로 test data로 뽑아내는 분할 방법입니다.
 - time split : 유저가 마지막으로 리뷰한 데이터를 test data로 뽑아내는 분할 방법입니다.
 - 리뷰 데이터가 5개 이하인 유저의 기록을 삭제합니다.
- Database
 - 처음에 시도한 GCP mysql은 작업비용 문제가 있었습니다.
 - mysql을 로컬에서 서버 배포를 진행하여 이 문제를 해결할 수 있었습니다.
 - 하지만 여전히 서버에 연결하는 시간 문제는 해결하지 못했습니다.
 - 그래서 서버가 필요 없는 SQLite를 사용하게 되었고 시간이 오래 걸리는 문제까지 해결할 수 있었습니다.

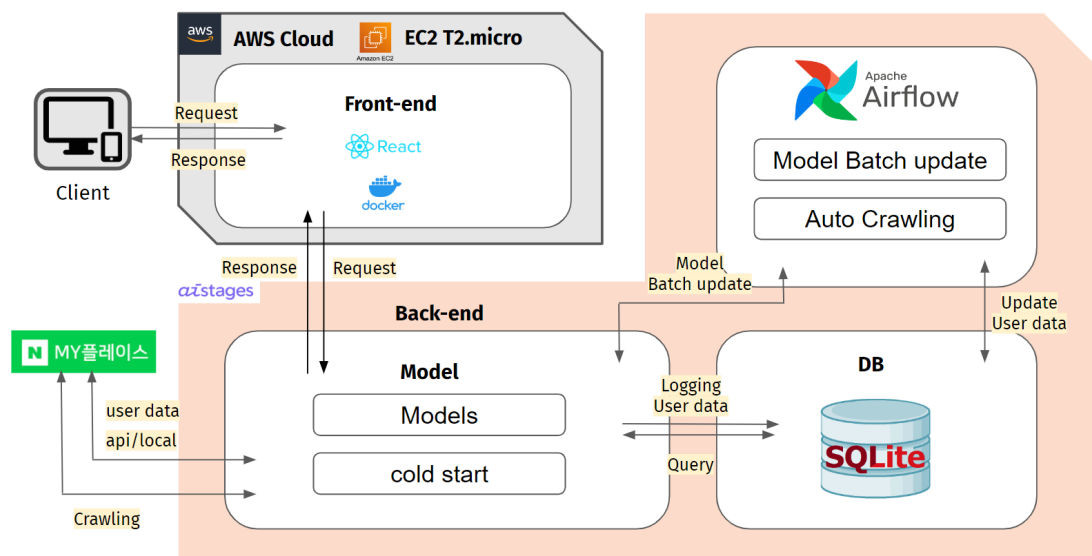
1-3-2. MODEL

- Models
 - SasRec
 - 시퀀셜한 추천을 위해 transformer를 이용한 SASRec 모델을 사용했습니다.

- MultiVAE
 - 생성모델인 VAE에 multinomial likelihood 개념을 도입한 MultiVAE 모델을 이용하여 implicit feedback 정보를 효과적으로 반영했습니다.
- EASE
 - autoencoder와 neighborhood-based 접근법을 융합한 단순한 구조의 EASE 모델을 사용했습니다.
- Offline test metric
 - recall@K : 이용자의 방문기록을 잘 맞추는지 평가하기 위함
 - personalization : 얼마나 개인화된 추천을 했는지 평가하기 위함
- Offline test 결과

model	recall@20 - random	recall@20 - time	Personalize
SASRec	5.65%	5.96%	0.00669
MultiVAE	11.23%	10.02%	0.00253
EASE	29.10%	24.29%	0.00334
단순 random	0.01%	0.01%	
단순 인기도	0.03%	0.03%	

1-3-3. PRODUCT SERVING



- 전체 구조 : aws ec2에 올라간 front-end와 v100 서버에 올린 back-end로 구성

- 진행 과정

1. 사용자가 aws 서버에 접속
2. 유저의 id 정보와 추천 장소, 식사 종류를 선택하여 넘김
3. 백엔드에서 해당 데이터를 db에서 확인
4. cold start 유저인지 아닌지 판단하여 다르게 추천을 진행
 - 새로운 유저
 - 리뷰 데이터가 5개 이하인 유저
 - 음식 호불호 테스트를 진행 → 인기도 기반 모델을 사용한 추천
 - 기존의 유저
 - 리뷰 데이터가 6개 이상인 유저
 - SASRec, MultiVAE, EASE의 딥러닝 모델을 통해 추천
5. 추천 결과에 대한 유저의 긍정/ 부정 클릭 정보는 백엔드로 보내져 DB에 저장
 - 이 정보는 후에 모델의 batch update를 하는데 사용

- A/B test



- “식당 링크 열기” 버튼
 - 해당 식당에 대한 긍정 피드백으로 기록합니다.

- “다른 결과 보기” 버튼

- 현재 화면에 노출된 식당 3개 모두에 대한 부정 피드백으로 기록합니다.

- Airflow

- auto crawling

- 리뷰 데이터를 매일 크롤링하면서 데이터를 최신화합니다.
 - 수집된 최신 데이터는, airflow로 주기적으로 진행되는 모델 학습에 사용됩니다.

- auto learning

- 사용자 사용 이력을 수집해서 Positive, Negative 정보를 DB에 저장합니다.
 - 저장한 사용자 로그데이터를, airflow를 통해 batch 단위로 업데이트 하면서 모델을 재학습시킵니다.
 - 5개 이하의 리뷰를 가진 사용자가 로깅을 남긴다면 cold-start를 벗어나도록 합니다.
 - Negative 샘플링을 활용하는 모델의 경우, 랜덤한 Negative 샘플링 대신 유저의 Negative 피드백을 활용할 수 있기 때문에 모델 성능을 더욱 개선할 수 있었습니다.

- MLflow

- Airflow 내부 모델에 MLflow를 적용했습니다.
 - Airflow를 통해 모델 학습이 배치로 실행될 때마다, 각 학습에 사용된 변수와 하이퍼파라미터와 평가지표 및 모델을 MLflow로 저장하고 기록할 수 있습니다.

▼ 1-4. 자체 평가 의견

잘했던 점

- 개발 계획에 맞추어서 개발을 진행했습니다.
- MLOPs를 경험할 수 있는 프로젝트였습니다.
- 논리적인 구조를 이루기 위해 노력했고 어느정도 성공했습니다.
- 모델 성능이 유의미하게 나왔습니다.

시도 했으나 잘 되지 않았던 것들

- 하둡을 사용해서 데이터베이스를 구축하려 했지만 실패한 점이 아쉬움으로 남습니다.
- 빠르게 서비스를 하고 피드백을 받아보려 했으나 조금 늦어졌습니다.

아쉬웠던 점들

- 지역이 서울시로 한정된 부분이 아쉽습니다.
- UI가 조금 부족한 것이 아쉽습니다.
- Cold-Start 유저에게 더 다양한 추천을 하지 못한 것이 아쉽습니다.

▼ Part 2. 개인 회고

▼ 김성연

• 이번 프로젝트에서 나의 목표는 무엇이었는가?

- 1) 데이터 수집부터 전처리, 모델 학습 및 배포와 재학습 등의 MLOPs 과정을 경험해 보기.
- 2) 백엔드(FastAPI 등), DB 등 필요한 엔지니어링 작업을 경험해보고 익숙해지기
- 3) 더 적극적인 깃 사용 등으로 이전 프로젝트보다 더 좋은 팀 협업으로 좋은 결과 만들기.

• 나는 내 학습목표를 달성하기 위해 무엇을 어떻게 했는가?

잘 하는 부분을 많이 하기 보단 배워야 하는 부분을 하려고 노력했습니다. 모델링을 직접적으로 엔지니어링 하는 것 보다 DB 스키마 정의, 문제 정의 등 큰 틀을 보려고 노력했습니다. 또 FastAPI, AirFlow 등의 틀을 다루는 강의를 열심히 들었습니다. 이후 이 부분을 우리 프로젝트에 도입하였습니다. 마지막으로 팀원과의 소통 비중을 크게 하고 전반적인 팀 스케줄 관리를 하였습니다. Master, Develop 등의 깃 브랜치 전략을 적극적으로 활용했습니다.

• 나는 어떤 방식으로 모델을 개선했는가?

직접적으로 모델을 개선하진 않았습니다. 다만 데이터를 정제하고 Train과 Test로 분리하여 모델 학습할 수 있는 환경을 만들었습니다. 이때 Time과 Rand 방식으로 분리하여 Test 데이터 셋의 성질을 다르게 분류하여 여러가지 관점으로 바라보도록 하였습니다.

실제 사용자의 만족도를 표현하는 Metric은 어떤 것일지 지속적으로 고민했습니다. 유저의 방문 이력을 맞추는 Recall@K와 얼마나 다양한 추천을 했는지를 나타내는 Personalization을 구할 수 있는 베이스라인을 SASRec 모델을 이용해 만들었습니다. 팀원들이 모델링에 집중할 수 있는 환경을 만들었습니다.

- **전과 비교해서, 내가 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?**

모델링은 많이 해봤지만 데이터 수집부터 서비스까지 End-to-End는 처음 해보지만 공부하기 정말 좋은 환경이기 때문에 배우자는 마인드로 시작했던 것 같습니다.

- **마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?**

진행이 생각보다 딜레이되는데 계속 공부를 하고 있다는 압박감, 이미 힘든 일정을 소화했는데 익숙하지 않은 툴을 다룰 때의 어려움이 배워보자는 열의를 많이 꺾었습니다. 되돌아보니 마음이 너무 급했던 것 같아요. 많이 배울 수 있었는데 처음 각오만큼 못 배운 것이 아쉽습니다.

- **한계/교훈을 바탕으로 다음 프로젝트에서 스스로 새롭게 시도해볼 것은 무엇일까?**

- 1) FastAPI와 AirFlow 다양하게 실습해보며 내 것으로 만들기.
- 2) 수집한 데이터로 추천 시스템 모델 더 공부해보기.

▼ 배성재

- **이번 프로젝트에서 나의 목표는 무엇이였는가?**

1. 실제 유저들의 데이터 수집부터 추천, 서비스 배포를 통해 실제 환경에서 가능한 추천 시스템의 전체 사이클을 경험하는 것
2. 유저에게 설명할 수 있는 추천 시스템을 만들기

- **나는 내 학습목표를 달성하기 위해 무엇을 어떻게 했는가?**

데이터 수집부터 모델 학습, 서비스 배포까지 모든 절차를 수행하기 위해서는 모델 외에도 데이터 수집, 웹 페이지 개발, 서비스 환경 구축이 필요했습니다. 이번 프로젝트에서 데이터 수집, 웹 페이지 개발, 서비스 환경 구축 개발을 주로 진행했습니다.

유저에게 설명가능한 추천시스템을 만들기 위해 팀원들이 만들어준 모델 결과에 설명을 첨부하는 과정을 진행했습니다. 프로젝트 서빙까지의 시간 부족으로 nlp를 기반으로 한 유저 리뷰 데이터 감성 분석과 추천의 이유를 타당하게 설명하는 과정을 진행하지 못한 점이 아쉬움으로 남습니다.

- **전과 비교해서, 내가 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?**

추천 모델 개발 외에 전체 서비스 배포를 주로 생각하며 개발했습니다. 추천 모델 학습 외에도 추천 시스템을 추천하기 위해서는 모델 추천 결과와 비슷하게 다양한 개발 과정을 경험했습니다.

- **마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?**

주어진 시간 안에 배포를 경험해보기 위해 웹 페이지와 서버 개발에 시간을 소비되어 스스로 고민했던 점들을 프로젝트에 적용하지 못한 점이 아쉬움으로 남습니다. 한정된 기간 안에 개발을 진행하기 위해서 타협했던 부분들이 아쉬움으로 남습니다.

- **한계/교훈을 바탕으로 다음 프로젝트에서 스스로 새롭게 시도해볼 것은 무엇일까?**

- 1) 프로젝트에 적용된 airflow 실습과 실험
- 2) 현재 데이터를 기반으로 추천 모델 공부

▼ 양승훈

- **이번 프로젝트에서 나의 목표는 무엇이었는가?**

- 데이터베이스에 유저 피드백 데이터를 적재하고, 주기적으로 모델을 최신화 해보기
- 다양한 평가 지표 적용하고, 모델 별로 비교
- 온라인 테스트를 실행할 수 있도록 서비스 기획

- **나는 내 학습목표를 달성하기 위해 무엇을 어떻게 했는가?**

- 유저 피드백을 잘 반영할 수 있도록 서비스 기획

- **나는 어떤 방식으로 모델을 개선했는가?**

- 유저 피드백을 통한 모델 업데이트를 통해 모델 개선

- **전과 비교해서, 내가 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?**

- 모델 개선 부분에서, 하이퍼파라미터 튜닝보다 유저 피드백에 더 집중
 - 유저 친화적인 서비스를 개발하는 데 도움됨

- **마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?**

- 유저 피드백 / 평가 지표 / 온라인 테스트 등이 이번 프로젝트의 목표였는데, 프로젝트를 진행하다 보니 기술적인 구현에 더 힘이 들어간 것 같음
- 데이터 수집을 크롤링에 의존하다 보니, 누락된 데이터가 상당수 존재하고 서울 지역에 한정됨

- **한계/교훈을 바탕으로 다음 프로젝트에서 스스로 새롭게 시도해볼 것은 무엇일까?**

- 더 체계적인 유저 피드백 시스템 구축
- 다양한 평가 지표 적용하고 모델 간 비교해보기

▼ 조수연

- **이번 프로젝트에서 나의 목표는 무엇이었는가?**

하나의 프로젝트가 전체적으로 어떻게 흘러가는지 파악하는 것이 목표였습니다.

- **나는 내 학습목표를 달성하기 위해 무엇을 어떻게 했는가?**

프로덕트 서빙에 관해서는 전혀 지식이 없었지만 강의와 실습내용을 바탕으로 코드를 직접 작성해 보고, 팀원들에게 안 되는 부분들을 질문해가며 배웠습니다.

- **전과 비교해서, 내가 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?**

이전에는 대회라는 것에 매몰되어 강의를 대충 학습하고 성급하게 무언가를 해 보려는 경향이 있었습니다. 그러나 이번 프로젝트에서는 우선 강의 내용과 팀원 간 논의한 내용 등을 먼저 이해하려고 노력하다 보니 프로젝트의 전체적인 구조를 어느 정도 파악할 수 있었다고 생각합니다.

- **마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?**

프로그래밍 기초가 부족해서 원하는 대로 코드를 작성하는 것, 프로젝트 아키텍처에 대한 이해 등이 부족했습니다.

- **한계/교훈을 바탕으로 다음 프로젝트에서 스스로 새롭게 시도해볼 것은 무엇일까?**

이번 프로젝트에서 프로그래밍 기초가 부족한 탓에 많은 작업을 할 수 없었다는 사실을 깨닫고, 프로젝트의 전체적인 구조를 이해할 수 있었습니다. 따라서 앞으로 기본 실력을 충분히 다져놓는다면 원활하게 프로젝트를 완성시킬 수 있겠다고 생각합니다. 또한 실제 서비스를 만든다는 것은 추천시스템이라는 도메인에 한해서만 생각할

것이 아니라 CV, NLP 등 다른 부분도 알고 있다면 훨씬 더 가치있는 프로젝트를 만들 수 있다는 것을 깨달았습니다.

▼ 황선태

- 이번 프로젝트에서 나의 목표는 무엇이었는가?

데이터 크롤링을 무사히 마치고 데이터를 분석하여 적절한 추천을 해주는 딥러닝 모델을 설계하는 것.

- 나는 어떤 방식으로 모델을 개선했는가?

먼저 EASE 모델을 이번 프로젝트의 데이터에 맞게 적용시키는 작업을 했다. 그리고 Multi-VAE를 이용한 추천 모델을 만들었다. 두 모델이 이번 프로젝트에서 0.29, 0.10이라는 높은 성능을 보였고, 이를 이번 프로젝트의 방대한 데이터가 들어갔을 때에도 돌아갈 수 있게 하는 일을 했다

- 전과 비교해서, 내가 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

모델을 데이터에 맞게 수정하고 또 튜닝을 통해 발전시켰으며, Airflow에 적용시키기 위해서 변형시키는 일을 하면서 차원을 맞추는 일을 했다. 이 과정에서 파이토치를 다루는 능력이 전반적으로 향상 될 수 있었다.

- 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

내가 사용한 모델들이 전부 user-item 의 관계만을 이용한 예측 모델이었다. 물론 오프라인 지표 성능은 만족스럽게 나왔지만, 식당 카테고리, 리뷰데이터와 같은 추가적인 데이터를 이용한 추천을 해줄 수 있는 모델을 만들어내지 못해 아쉽다.

모델을 직접 작성하는 것은 상당히 힘든 문제였다. 데이터 전처리, 제출파일 만들기 와 같은 기본적인 일들이 생각보다 힘들다는 것을 느꼈고 이를 혼자서 해결하려고 하기 보다는 검색과 동료를 적극적으로 활용하는게 맞다는 생각을 하게 되었다.

- 한계/교훈을 바탕으로 다음 프로젝트에서 스스로 새롭게 시도해볼 것은 무엇일까?

다음 프로젝트를 하기 위해서 NLP 공부가 필수적이라는 것을 느꼈다. NLP 공부를 통해 시퀀셜 데이터들을 다룰 수 있는 능력을 갖출 것이다.

▼ 홍재형

- 이번 프로젝트에서 나의 목표는 무엇이었는가?

Product serving 파트 강의에 CI/CD, Fastapi, Docker, Airflow 등등 많은 내용들이 있었는데 모든 기술을 프로젝트에 적용시켜 보는 것이 목표였습니다.

- 나는 내 학습목표를 달성하기 위해 무엇을 어떻게 했는가?

처음 역할을 나누어서 프로젝트를 진행했는데 다른 팀원들이 작성한 코드를 읽어두고 나중에 유지보수를 할 때 도와서 진행을 했습니다.

- **나는 어떤 방식으로 프로젝트를 개선했는가?**

데이터베이스 선정에 고민을 많이 했었고 Mysql(GCP), Mysql(local), SQLite를 실험해 봤습니다. GCP와 Local에서 mysql끼리 비교를 했을 때 속도는 비슷했지만 GCP는 돈이 든다는 단점이 있었고 Local에서는 컴퓨터를 상시 켜둬야 한다는 단점이 있었습니다. 공통적인 단점으로는 서버에 연결해야 하는 구조였고 이 부분 때문에 시간소요가 있었습니다. 그래서 SQLite를 사용했고 시간을 3배 정도 단축시킬 수 있었습니다. 또한 앞서 말한 단점들도 모두 해결할 수 있었습니다.

- **전과 비교해서, 내가 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?**

이전 프로젝트에서는 다양한 모델을 사용해 보고 모델을 개선하는 데에 집중을 했었는데 이번에는 데이터를 수집을 자동화하고 만들어진 모델을 배포하는 것에 집중했습니다.

- **마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?**

사실 데이터베이스를 처음에는 하둡을 사용하려고 했었습니다. 로컬에서 하둡서버를 만들었고 GCP에서도 구축을 했었으나 주어진 시간 안에 프로젝트를 진행하는데 발목이 잡힐 것 같아 포기했던 점이 아쉬웠습니다.

- **한계/교훈을 바탕으로 다음 프로젝트에서 스스로 새롭게 시도해볼 것은 무엇일까?**

시간이 부족할 것 같아 하둡을 하다가 말았었는데 다음에는 조금 더 데이터를 많이 모은 다음 하둡을 통해 빅데이터를 다뤄보고 싶습니다.