

MLOps 프로젝트

영화 평점 예측 서비스 개발 및 MLOps 파이프라인 구축

목차

01

주제 선정

02

전체 아키텍처

03

기술적 내용 및 해결방법

04

소감 및 향후 계획

주제 선정

TMDB API 기반

영화 평점 예측 시스템 구축

01

로컬 개발 모델을 API로 배포하고 MLOps 파이프라인 일부 구현.
모델 코드 모듈화, 컨테이너 기반 학습 환경 및 API배포 구현

02

인기 영화 데이터를 자동으로 수집하고 학습하는 파이프라인
경험.

데이터 수집 (Airflow) - 서빙 (FastAPI) - 사용자 UI (Streamlit)
까지의 End-to-End 시스템 구현

03

개봉 전 영화의 줄거리 감성 분석을 통해 평점을 예측하는
비즈니스 가치 창출

아키텍처



아키텍처

인스턴스 A

Step1: Automated pipeline 모델 학습 및 업로드

- 수집: TMDB API를 통해 매일 100개 데이터 수집
- 전처리: TextBlob라이브러리 활용해 줄거리에서 감성 점수 (sentiment)와 길이(len) 피처를 추출하고, 장르, 언어에 적용된 LabelEncoder매칭값 저장
- 학습: XGBoost, LightGBM, Random Forest중 선택모델 학습
- 실험 관리: Wandb에 모델, 하이퍼파라미터 조합의 성능 기록
- 아티팩트:날짜별 전처리 파일, 날짜별 모델, best_score, 베스트 모델 파일 업데이트. 과적합 방지를 위해 $rmse_gap(=train_rmse-test_rmse)$ 이 직전 값보다 작으면 베스트 모델로 갱신하고 .pkl 파일로 저장
- 모델 업로드:베스트 모델과 전처리에 사용된 인코더 .pkl 파일을 S3로 저장

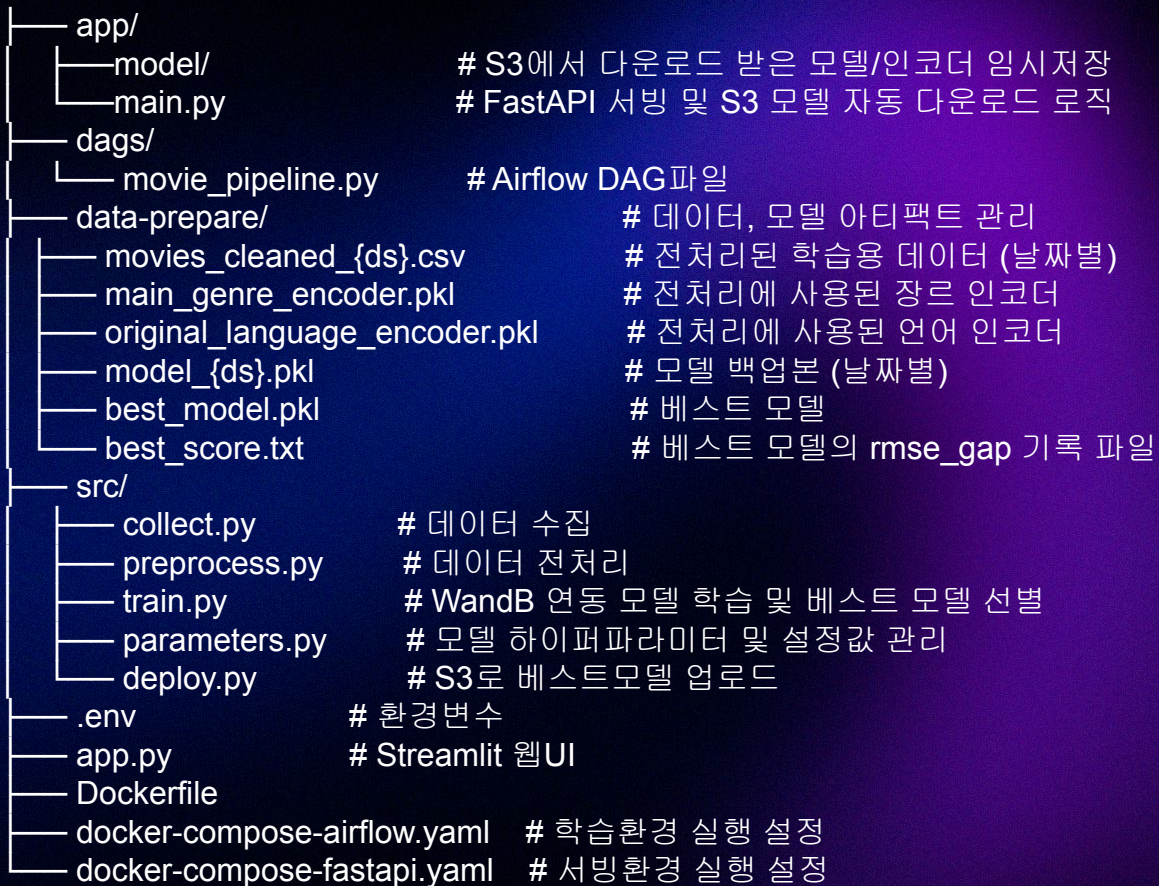
Step2: 저장소

인스턴스 B

Step3: Serving pipeline 실시간 모델 추론 및 서비스

- 서빙: FastAPI 서버 시작 시 S3에서 최신 모델 자동 다운로드. 예측 API 제공.
- UI: 사용자가 영화 정보를 입력하고 예상 평점 확인. deep-translator를 통해 한글 줄거리를 영문으로 번역하여 모델이 인식. 번역된 텍스트의 감성을 분석.

아키텍처



기술적 내용 및 해결방법 #1

1. 시스템 가용성 (Scale-out)

2. 데이터 신뢰성 (Data Leakage)
3. 실험 관리 (Wandb)
4. 아키텍처 고도화 (S3)
5. 데이터 일관성 (Encoding)

문제

단일 t3.small 인스턴스에서 Airflow와 Docker 컨테이너들을
동시 가동 시 RAM 부족으로 인한 시스템 다운 발생

해결

Scale-up 대신 Scale-out 방식을 채택하여 학습(Airflow)와
서빙(FastAPI) 인스턴스를 분리하고 부하를 분산함.

리소스 간섭을 방지하고 분산 환경에서의 안정적인 시스템
운영 역량을 확보함

기술적 내용 및 해결방법 #1

분산 환경 (Scale-out) : Airflow, FastAPI

인스턴스 (3) 정보

인스턴스를 속성 또는 (case-sensitive) 태그로 찾기

모든 상태 ▼

<input type="checkbox"/>	Name	인스턴스 ID	인스턴스 상태	인스턴스 유형	상태 검사	경보 상태	가용 영역
<input type="checkbox"/>	mlops-user2		실행 중	t3.micro	3/3개 검사 통과	경보 보기 +	ap-northeast-2c
<input type="checkbox"/>	mlops-user1		실행 중	t3.small	3/3개 검사 통과	경보 보기 +	ap-northeast-2c

```

✓ Network airflow_mlops_default Removed
0.1s
ubuntu@ip-172-31-44-5:~/airflow_MLOps$ docker compose -f docker-compose-airflow.yaml up -d
[+] up 5/5
✓ Network airflow_mlops_default Created
0.1s
✓ Container airflow_mlops-postgres-1 Healthy
6.0s
✓ Container airflow_mlops-airflow-scheduler-1 Created
0.1s
✓ Container airflow_mlops-airflow-init-1 Created
0.1s
✓ Container airflow_mlops-airflow-webserver-1 Created
0.1s
ubuntu@ip-172-31-44-5:~/airflow_MLOps$

[+] up 3/3
✓ Image airflow_mlops-fastapi Built 153.4s
✓ Network airflow_mlops_default Created 0.1s
✓ Container airflow_mlops-fastapi-1 Created 0.2s
ubuntu@ip-172-31-32-66:~/airflow_MLOps$ ls
Dockerfile app.py data-prep docker-compose-airflow.yaml mlops-user2.pem scripts
pycache app.py data-prep docker-compose-fastapi.yaml requirements.txt wandb
ubuntu@ip-172-31-32-66:~/airflow_MLOps$ cd ap
-bash: cd: ap: No such file or directory
ubuntu@ip-172-31-32-66:~/airflow_MLOps$ cd app
ubuntu@ip-172-31-32-66:~/airflow_MLOps/app$ ls
__pycache__ main.py model
ubuntu@ip-172-31-32-66:~/airflow_MLOps/app$ cd model
ubuntu@ip-172-31-32-66:~/airflow_MLOps/app/model$ ls
ubuntu@ip-172-31-32-66:~/airflow_MLOps/app/model$ cd ..
ubuntu@ip-172-31-32-66:~/airflow_MLOps/app$ cd ..
ubuntu@ip-172-31-32-66:~/airflow_MLOps$ docker ps
CONTAINER ID IMAGE COMMAND NAMES CREATED STATUS PORTS
55cb61d39d58 airflow_mlops-fastapi "/usr/bin/dumb-init ..." 3 minutes ago Up 3 minutes 0.0.0.0
:8000->8000/tcp, [::]:8000->8000/tcp, 8888/tcp airflow_mlops-fastapi-1
ubuntu@ip-172-31-32-66:~/airflow_MLOps$

```

기술적 내용 및 해결방법 #2

1. 시스템 가용성 (Scale-out)

2. 데이터 신뢰성 (Data Leakage)

3. 실험 관리 (Wandb)

4. 아키텍처 고도화 (S3)

5. 데이터 일관성 (Encoding)

문제

초기 RMSE 0.44라는 극단적인 고성능 기록. EDA 분석 결과 미래 시점 데이터인 revenue, vote_count 변수가 포함된 Data Leakage 확인

해결

Leakage 변수를 식별하여 제거하고, overview(줄거리)의 감성점수(Sentiment), 길이(Length) 파생변수 생성하여 모델 일반화 개선

기술적 내용 및 해결방법 #2

데이터 신뢰성 확보

	id	title	budget	revenue	runtime	vote_count	vote_average	release_year	genre_encoded	lang_encoded	overview_len	overview_sentiment
0	1306368	The Rip	100000000	0	113	575	7.046	2026	0	1	146	0.000000
1	1043197	Dust Bunny	0	928164	106	74	6.900	2025	0	1	212	0.200000
2	1242898	Predator: Badlands	105000000	184478861	107	1574	7.800	2025	0	1	158	-0.133333
3	83533	Avatar: Fire and Ash	350000000	1320010340	198	1548	7.340	2025	11	1	361	-0.380519
4	1242501	Icefall	0	0	99	89	6.500	2025	0	1	379	-0.233333

1. 수집된 변수 26개:

```
['adult', 'backdrop_path', 'belongs_to_collection', 'budget', 'genres', 'homepage', 'id', 'imdb_id', 'origin_country', 'original_language', 'original_title', 'overview', 'popularity', 'poster_path', 'production_companies', 'production_countries', 'release_date', 'revenue', 'runtime', 'spoken_languages', 'status', 'tagline', 'title', 'video', 'vote_average', 'vote_count']
```

2. 전처리된 변수 12개:

```
['id', 'title', 'budget', 'revenue', 'runtime', 'vote_count', 'vote_average', 'release_year', 'genre_encoded', 'lang_encoded', 'overview_len', 'overview_sentiment']
```

3. X 변수 5개: ['runtime', 'genre_encoded', 'lang_encoded', 'overview_sentiment', 'overview_len']

불필요 변수 제외: vote_count, revenue, budget, release_year

4. y 변수: "vote_average"

기술적 내용 및 해결방법 #3

1. 시스템 가용성 (Scale-out)

2. 데이터 신뢰성 (Data Leakage)

3. 실험 관리 (Wandb)

4. 아키텍처 고도화 (S3)

5. 데이터 일관성 (Encoding)

문제

반복되는 하이퍼파라미터 튜닝과 재학습 과정에서 성능 지표를 기록하는 것에 한계가 있고 모델간 비교가 어려움

해결

Wandb를 연동하여 학습 로그를 시각화하고 실험 이력을 대시보드로 확인하며 최적의 파라미터를 체계적으로 선별 가능.

실험의 재현성을 확보하고 데이터에 기반한 모델 개선 근거 정립.

효율적인 실험 관리 및 모델 최적화 가능

기술적 내용 및 해결방법 #3

Wandb 도입

Runs

Search runs

Filter

Group

Sort

New sweep

NAME

4 visualized

STATUS

CREATED *

RUNTIME

MODEL

TYPE

PARAMS.LEARN

PARAMS.MAX_ID

PARAMS.N_SIZE

PARAMS.RANDP

PARAMS.SUBSAM

RMSE_GAP

TEST_RMSE

TRAIN_RMSE

xgboost_2026-01-27_depth4

Finished

2d ago

4s

xgboost

0.05

4

200

42

0.7

0.9094

0.98007

0.074075

xgboost_2026-01-27_depth4

Finished

2d ago

2s

xgboost

0.05

4

200

42

0.7

0.86418

0.94448

0.080299

xgboost_2026-01-27_depth4

Finished

2d ago

3s

xgboost

0.05

4

200

42

0.7

0.87507

0.95378

0.078715

xgboost_2026-01-27_depth4

Finished

2d ago

4s

xgboost

0.05

4

200

42

0.7

0.85509

0.93081

0.075723

xgboost_2026-01-26_depth4

Finished

2d ago

4s

xgboost

0.05

4

200

42

0.7

0.85509

0.93081

0.075723

xgboost_2026-01-23_depth4

Finished

4d ago

2s

xgboost

0.05

4

200

42

0.7

0.93973

1.04952

0.089794

xgboost_2026-01-23_depth4

Finished

4d ago

44s

xgboost

0.05

4

200

42

0.7

0.73805

0.82134

0.081285

xgboost_2026-01-24_depth4

Finished

4d ago

44s

xgboost

0.05

4

200

42

0.7

0.73805

0.82134

0.081285

xgboost_2026-01-23_depth4

Finished

6d ago

3s

xgboost

0.05

4

200

42

0.7

0.80036

0.87927

0.078611

xgboost_2026-01-23_depth4

Finished

6d ago

1s

xgboost

0.05

4

200

42

0.7

0.88324

0.95751

0.074372

xgboost_2026-01-05_44

Finished

6d ago

3s

xgboost

0.05

4

200

42

0.7

-

0.87415

-

xgboost_2026-01-23_depth4

Finished

6d ago

2s

xgboost

0.05

4

200

42

0.7

0.8924

0.96735

0.074953

xgboost_2026-01-24_depth4

Finished

7d ago

3s

xgboost

0.05

4

200

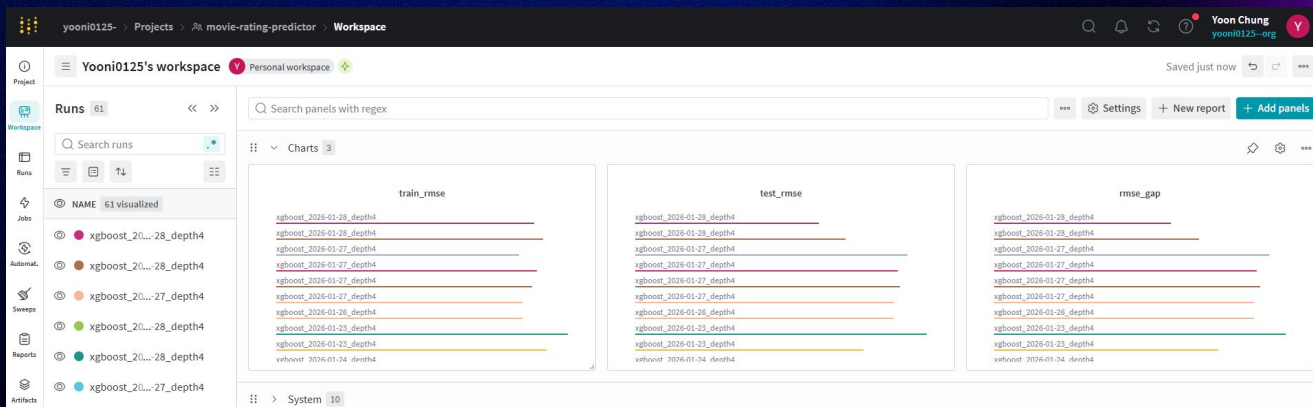
42

0.7

-

0.82486

-



기술적 내용 및 해결방법 #4~5

1. 시스템 가용성 (Scale-out)

2. 데이터 신뢰성 (Data Leakage)

3. 실험 관리 (Wandb)

4. 아키텍처 고도화 (S3)

5. 데이터 일관성 (Encoding)

문제

- 1) 실무 환경에서 인스턴스간 직접 전송(SCP)방식 위험성 인지
- 2) 학습과 서빙 시의 인코딩 매핑값이 달라져 예측 결과가 왜곡

해결

- 1) 인스턴스간 직접 전송 방식 폐기 → AWS S3 모델 레지스트리 활용한 모델 아티팩트 관리 방식 도입. 서빙 인스턴스와의 의존성 제거
- 2) LabelEncoder를 .pkl 파일로 저장하여 배포단계에 포함. 학습-검증-서빙 전 과정에서 데이터 일관성 확보

기술적 내용 및 해결방법 #4~5

S3 기반 모델 레지스트리

aws [검색] [알트+S] Amazon Q에게 질문하기

Amazon S3 > 버킷 > mlops-bucket-user1 > models/ > latest/

Amazon S3

- 버킷
 - 범용 버킷
 - 디렉터리 버킷
 - 테이블 버킷
 - 백터 버킷
- 액세스 관리 및 보안
 - 액세스 지점
 - FSx용 액세스 포인트
 - Access Grants
 - IAM Access Analyzer
- 스토리지 관리 및 인사이트
 - Storage Lens
 - 배치 작업

latest/

객체 속성

객체 (3) S3 URI 복사

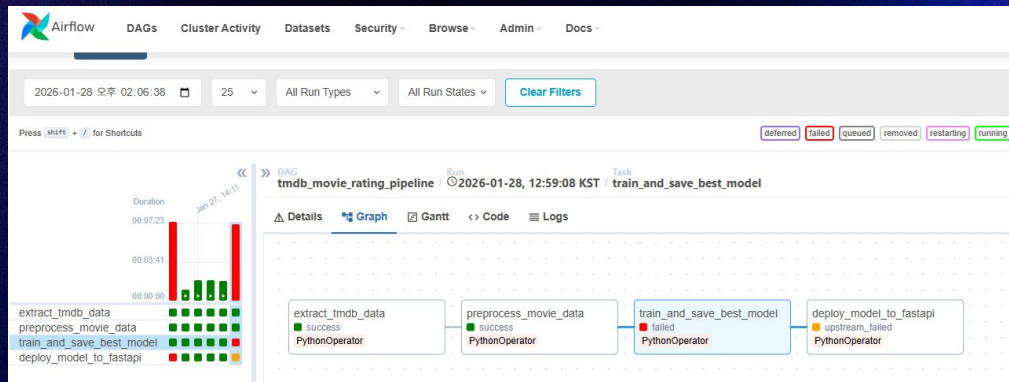
객체는 Amazon S3에 저장되어 있는 기본 엔터티입니다. [Amazon S3 인벤토리](#)를 사용하여 버킷에 있는 모든 객체의 목록을 얻을 수 있습니다.

Q 점두사로 객체 찾기

<input type="checkbox"/>	이름	▲	유형	▼	마지막 수정	▼	크기	▼	스토리지 클래스
<input type="checkbox"/>	best_model.pkl		pkl		2026. 1. 28. pm 5:29:54 PM KST		279.1KB		Standard
<input type="checkbox"/>	main_genre_encoder.pkl		pkl		2026. 1. 28. pm 5:29:54 PM KST		730.0B		Standard
<input type="checkbox"/>	original_language_encoder.pkl		pkl		2026. 1. 28. pm 5:29:54 PM KST		663.0B		Standard

프로젝트 산출물

Airflow



Clear and Retry

Task: deploy_model_to_fastapi
Run: manual__2026-01-27T03:59:39.346433+00:00

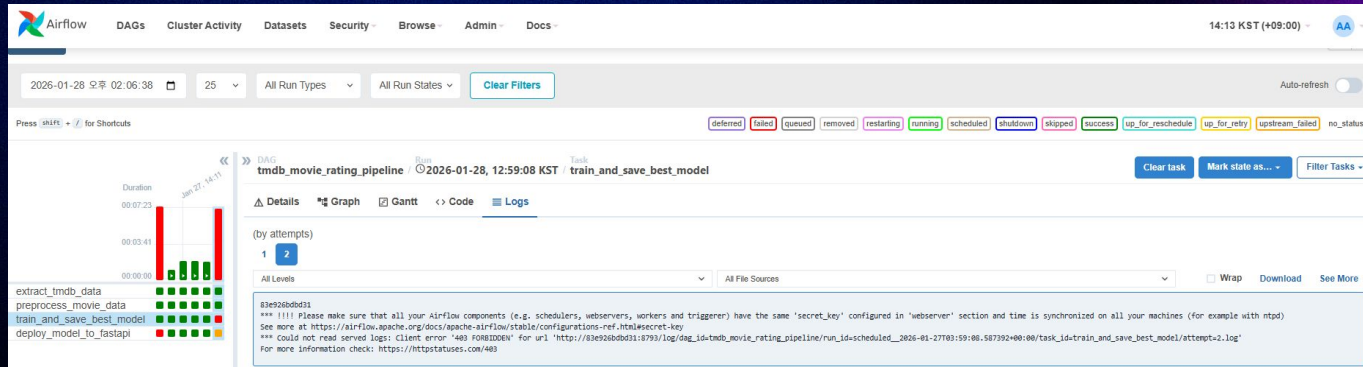
Include:

Past Future Upstream Downstream Recursive Only Failed

Affected Tasks: 1

TASK NAME	MAP INDEX	RUN ID
deploy_model_to_fastapi	-1	manual__2026-01-27T03:59:39.346433+00:00

Cancel Clear



프로젝트 산출물

FastAPI

FastAPI 0.1.0 GAS 5.1

default

GET /predict Predict

Parameters

Name Description

runtime * required
integer (query) 100

genre_encoded * required
integer (query) 1

lang_encoded * required
integer (query) 1

sentiment * required
number (query) -0.2

overview_len * required
integer (query) 22

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'http://13.125.86.24:8000/predict?runtime=100&genre_encoded=1&lang_encoded=1&sentiment=-0.2&overview_len=22' \
  -H 'accept: application/json'
```

Request URL

```
http://13.125.86.24:8000/predict?runtime=100&genre_encoded=1&lang_encoded=1&sentiment=-0.2&overview_len=22
```

Server response

Code Details

200

Response body

```
{
  "predicted_rating": 7.381941165433129
}
```

Response headers

```
server: uvicorn
content-type: application/json
date: Mon, 04 May 2020 09:40:00 GMT
server: uvicorn
```

Responses

Code Description Links

200 Successful Response No links

Media type
application/json

Controls: Accept header:

Example Value | Schema

```
"string"
```

422 Validation Error No links

Media type
application/json

Example Value | Schema

```
{
  "detail": [
    {
      "loc": [
        "string",
        0
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

프로젝트 산출물

Streamlit 기반 UI 설계

AI 영화 평점 예측기 (다국어 지원)

입력하신 영화 정보를 바탕으로 약 70-80% 정확도로 예상 평점을 분석합니다.

줄거리의 분위기(감성)까지 점수에 반영됩니다!

상영시간(분)

120

- +

언어

de

▼

주요 장르

Action

▼

영화 줄거리

이곳에 영화의 줄거리를 입력하세요. 길이에 따라 예측값이 달라집니다.
한글로 입력하셔도 AI가 번역하여 분석합니다.

예상 평점 확인

AI 영화 평점 예측기 (다국어 지원)

입력하신 영화 정보를 바탕으로 약 70-80% 정확도로 예상 평점을 분석합니다.

줄거리의 분위기(감성)까지 점수에 반영됩니다!

상영시간(분)

103

- +

언어

ko

▼

주요 장르

Drama

▼

영화 줄거리

살아있는 기적이라 불리는 시각장애인 전각 장인 '임영규'의 아들 '임동환'이 40년 전 실종된 줄 알았던 어머니의 백골 시신 발견 후, 그 죽을 뒤의 진실을 파헤치는 이야기

예상 평점 확인

분석 결과

예상 평점

6.47 / 10

줄거리 분위기

어둡고/부정적 ● (점수: -0.37)

💡 이 시뮬레이션은 'Drama' 장르와 'ko' 언어의 특성을 반영한 결과입니다.

프로젝트 산출물

프로젝트 관리 (Notion, Git)

MLOps 프로젝트						
TMDB 데이터를 활용한 영화 평점 예측 서비스 개발 및 MLOps 파이프라인 구축						
작업	상태	담당자	마감일	우선순위	설명	
프로젝트 기초 공부	완료		01/21/2026	중간	프로젝트 진행 전 MLOps 공부, 참고자료	
주제 및 방향 설정	완료		01/21/2026	중간	프로젝트 주제 및 업무 분장	
Data 수집	완료		01/21/2026	높음	데이터 크롤링 진행 # Data source : https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset # ratings_small.csv: The subset of 100,000 ratings from 700 users on 9,000 movies	
1차 엔토링	완료		01/22/2026		설계 점검	
전체 아키텍처 설계	진행 중		01/23/2026	높음	엔토링 전, 사전 아키텍처 설계	
Data 전처리	완료		01/23/2026	높음	이상치, 결측치, 인코딩, 데이터 누수, 유효성 분석	
모델링	완료		01/23/2026	높음	LightGBM, RandomForest, XGBoost 등 모델 선택, 파라미터 선택	
모델 배포	진행 중		01/26/2026	중간	배포 UI, API 배포	
Automated pipeline 구축	진행 중		01/26/2026	높음	성능 평가 방식에 따른 베스트 모델 선정, 저장	
2차 엔토링	시작 전		01/26/2026		구현 중간 점검	
Serving pipeline 구축	진행 중		01/27/2026	높음	모델 추론에 따른 결과값 서빙	
HTML 구축	시작 전		01/27/2026	중간	결과값 출력물 HTML로 구축	
EC2 서버 구축 및 환경 설정	시작 전		01/28/2026	낮음		
EC2 서버에 모든 파이썬 인코딩/테스팅	시작 전		01/28/2026	높음	테스팅	
3차 엔토링	시작 전		01/28/2026		완성도 점검	
PPT 작성	시작 전		01/29/2026	중간	발표자료 작성	

feat/add language_encoder #4
 Jun-Yoon merged 1 commit into main from scripts 3 hours ago

Conversation
 Commits 1
Checks 0
Files changed 1

yoon-chung commented yesterday
 Collaborator

무엇을 했나요?

-original_language 필드 추가, language_encoder 추가

왜 필요한가요?

-모델 성능 개선

테스트 방법

- [이] 로컬 실행 확인
- [이] 불필요한 코드 제거

체크리스트

- [] 로컬 실행 확인
- [] 불필요한 코드 제거

Reviews

No reviews

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

None yet

feat/add language_encoder #4
 Jun-Yoon merged 1 commit into main from scripts 3 hours ago

Conversation
 Commits 1
Checks 0
Files changed 1

Changes from all commits
 File filter
Conversations
Jump to

feat/add language_encoder
 main (4)

yoon-chung committed yesterday

```

1  def preprocess_data(input_filenames='movies_manual.csv', output_filenames='no
17  df = pd.read_csv(input_path)
18  print(f"원본 데이터 크기: {df.shape}")
19
20  # cols = ['id', 'title', 'budget', 'revenue', 'runtime', 'genres', 'release_date', 'vote_count', 'vote_average']
21  # cols = ['id', 'title', 'original_language', 'budget', 'revenue', 'runtime', 'genres', 'release_date', 'vote_count', 'vote_average']
22  available_cols = [c for c in df.columns if c in df.columns]
23  df = df[available_cols]
24
25  def extract_genre(genre_str):
26  encoder_path = os.path.join(save_dir, "genre_encoder.pkl")
27  joblib.dump(le, encoder_path)
28
29  # 언어 인코딩 (Label Encoding)
30  le_lang = LabelEncoder()
31  df['lang_encoded'] = le_lang.fit_transform(df['original_language']).astype(str)
32
33  # 언어 인코딩 저장
34  encoder_path = os.path.join(save_dir, "language_encoder.pkl")
35  joblib.dump(le_lang, encoder_path)
36
37  # 날짜 데이터 처리 (연도 추출)
38  df['release_year'] = pd.to_datetime(df['release_date']).dt.year
39
40  # 불필요한 컬럼 삭제
41  df_clean = df.drop(['genres', 'release_date', 'main_genre'], axis=1)

```

소감 및 향후 계획

성장한 내용

모델 코드 모듈화와 컨테이너 기반 환경 구축, AWS, Docker를 활용하여 로컬 개발 환경을 클라우드에 구현하는 역량을 쌓았습니다.

특히 리소스 간섭 등 여러 기술적 문제를 해결하는 과정에서, DAG파일 편집, Docker 컨테이너 기반 기술, airflow workflow 구성, S3 연동 등 엔지니어링 기술에 조금 친숙해진 시간이었습니다.

또한 Wandb (Weights&Biases)를 활용한 실험 기록 관리를 통해 효율적인 하이퍼파라미터 튜닝 프로세스를 경험하였습니다.

향후 계획

본 프로젝트에서는 AWS S3 기반 모델 레지스트리를 구축하고 배포 파이프라인을 구현하며 MLOps 기틀을 마련했습니다. 향후 모델의 버전별 기록과 배포가 자동으로 관리되는 수준 높은 MLOps 환경을 경험해보고 싶습니다.

소스코드 <https://github.com/yoon-chung/MLOps.git>

감사합니다