

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

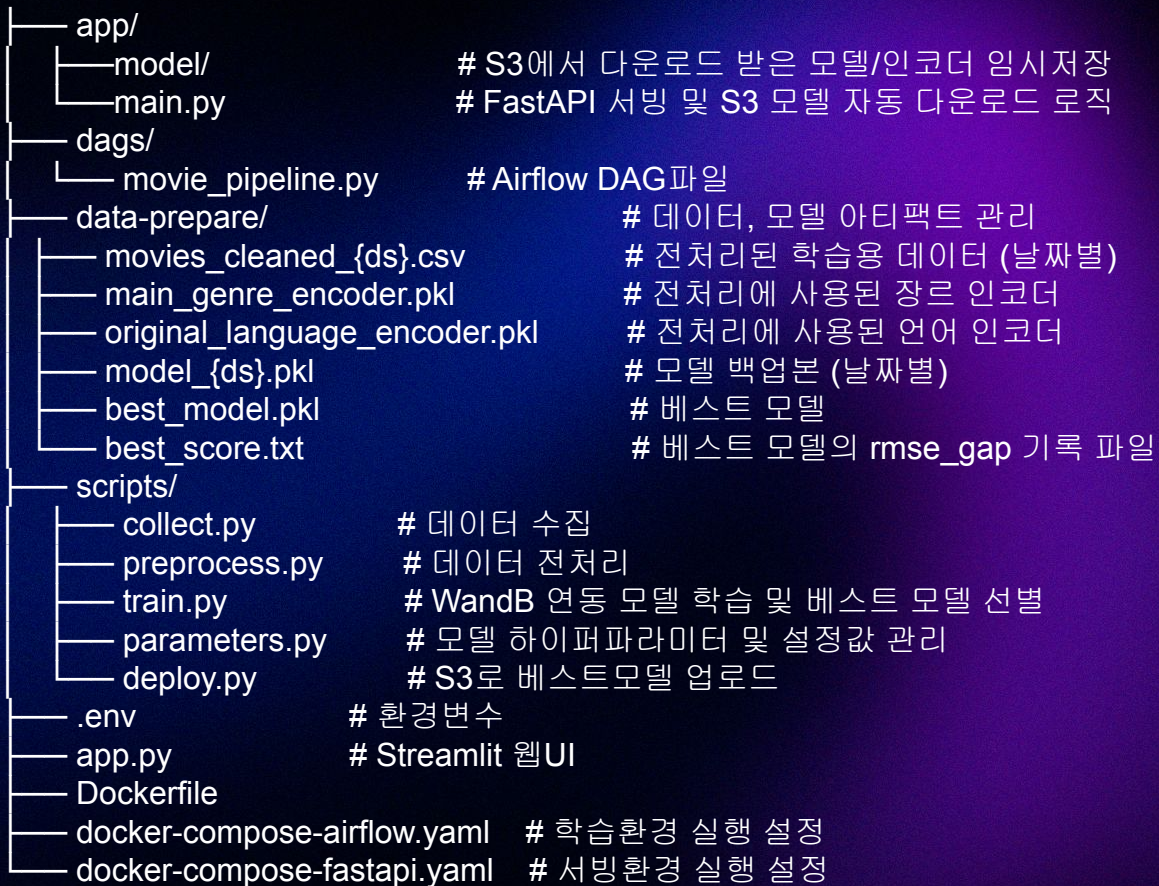
Step2: 저장소

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$ 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_RANDOM

PARAMS_SUBSET

RMSE_GAP

TEST_RMSE

TRAIN_RMSE

xgboost_2026-01-27_depth4

Finished

2d ago

4s

xgboost

0.05

4

200

42

0.7

0.9054

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

0.080295

xgboost_2026-01-27_depth4

Finished

2d ago

3s

xgboost

0.05

4

200

42

0.7

0.87507

0.95378

0.078115

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

0.87527

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

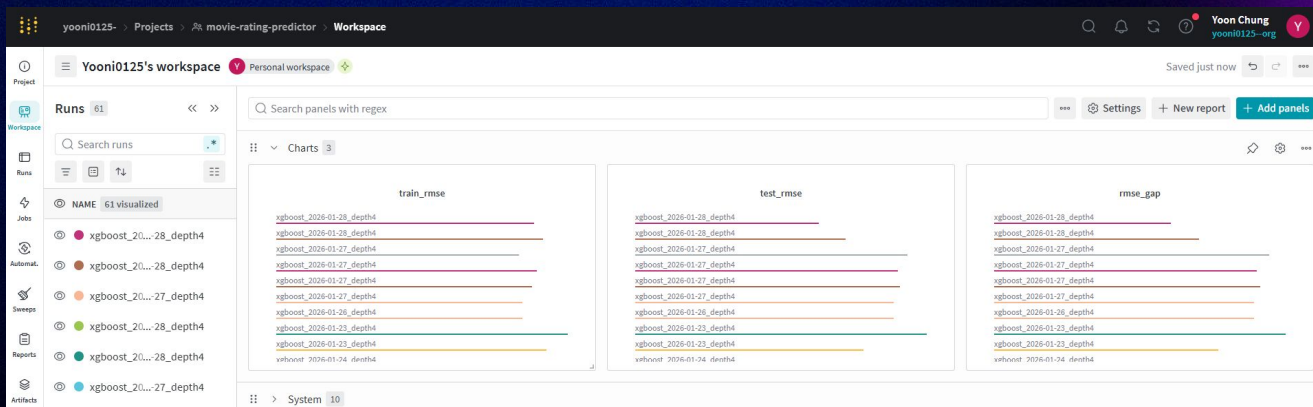
42

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

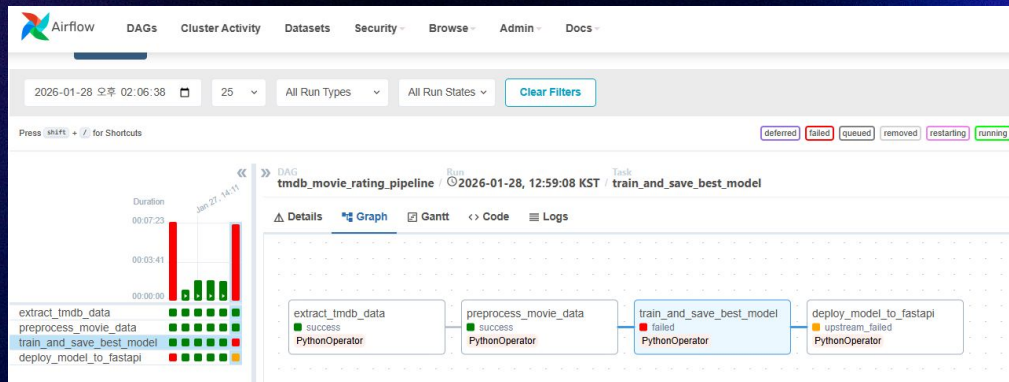
객체는 Amazon S3에 저장되어 있는 기본 엔터티입니다. [Amazon S3 인벤토리](#)를 사용하여 버킷에 있는 모든 객체의 목록을 얻을 수 있습니다.

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

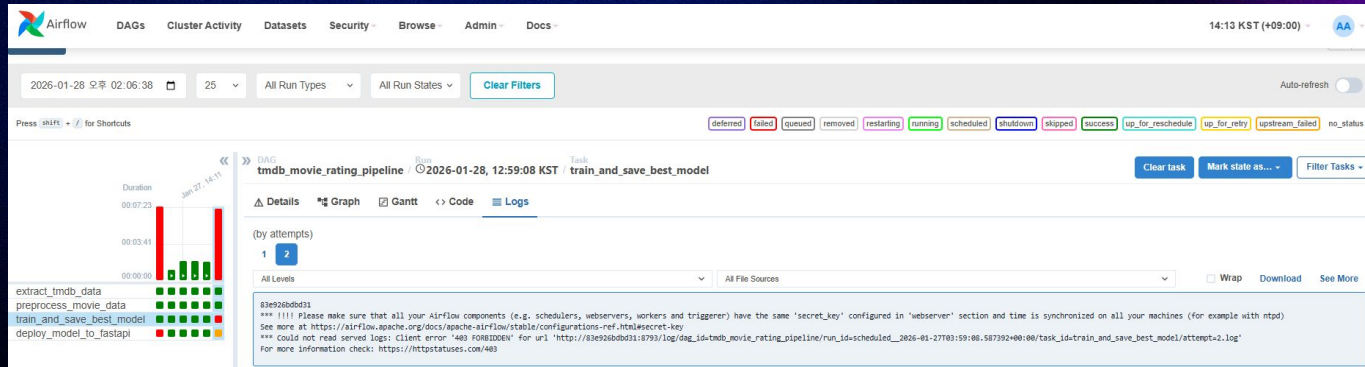
Include:

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

openapi.json

default

GET /predict Predict

Parameters

Cancel

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	<p>Response body</p> <pre>{ "predicted_rating": 7.345941585433129 }</pre> <p>Response headers</p> <pre>server: uvicorn content-type: application/json date: Mon, 04 May 2020 09:40:00 GMT access-control-allow-origin: *</pre>

Responses

Code	Description	Links
200	<p>Successful Response</p> <p>Media type: application/json</p> <p>Controls: Accept header:</p> <p>Example Value Schema</p> <pre>"string"</pre>	No links
422	<p>Validation Error</p> <p>Media type: application/json</p> <p>Example Value Schema</p> <pre>{ "detail": [{ "loc": ["string", 0], "msg": "string", "type": "string" }] }</pre>	No links

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)

MLOps프로젝트						
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 및 환경	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 출력용 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-Yeol merged 1 commit into main from scripts 3 hours ago

Conversation 0 Commits 1 Checks 0 Files changed 1

yoon-chung commented yesterday

무엇을 했나요?

-original_language 필드 추가, language_encoder 추가

왜 필요한가요?

-모델 성능 개선

테스트 방법

•

체크리스트

- [x] 로컬 실행 확인
- [x] 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-Yeol merged 1 commit into main from scripts 3 hours ago

Conversation 0 Commits 1 Checks 0 Files changed 1

Changes from all commits 1 File filter Conversations 1 Jump to

feat/add language_encoder

main (4)

yoon-chung committed yesterday

```

1  diff --git a/scripts/preprocess.py b/scripts/preprocess.py
2  index 11111..22222
3  --- a/scripts/preprocess.py
4  +++ b/scripts/preprocess.py
5  @@ -1,17 +1,17 @@
6  def preprocess_tmdb_data(input_filenames="movies_manual.csv", output_filenames="no
7  17 17 df = pd.read_csv(input_path)
8  18 18 print(f"원본 데이터 시작 - 원본 데이터 크기: {df.shape}")
9  19 19
10 20 - cols = ['id', 'title', 'budget', 'revenue', 'runtime', 'genres', 'release_date', 'vote_count', 'vote_average']
11 21 + cols = ['id', 'title', 'original_language', 'budget', 'revenue', 'runtime', 'genres', 'release_date', 'vote_count', 'vote_average']
12 22 + available_cols = [c for c in cols if c in df.columns]
13 23 df = df[available_cols]
14 24
15 25
16 26
17 27
18 28
19 29
20 30
21 31
22 32
23 33
24 34
25 35
26 36
27 37
28 38
29 39
30 40
31 41
32 42
33 43
34 44
35 45
36 46
37 47
38 48
39 49
40 50
41 51
42 52
43 53
44 54
45 55
46 56
47 57
48 58
49 59
50 60
51 61
52 62
53 63
54 64
55 65
56 66
57 67
58 68
59 69
60 70
61 71
62 72
63 73
64 74
65 75
66 76
67 77
68 78
69 79
70 80
71 81
72 82
73 83
74 84
75 85
76 86
77 87
78 88
79 89
80 90
81 91
82 92
83 93
84 94
85 95
86 96
87 97
88 98
89 99
90 100
91 101
92 102
93 103
94 104
95 105
96 106
97 107
98 108
99 109
100 110
101 111
102 112
103 113
104 114
105 115
106 116
107 117
108 118
109 119
110 120
111 121
112 122
113 123
114 124
115 125
116 126
117 127
118 128
119 129
120 130
121 131
122 132
123 133
124 134
125 135
126 136
127 137
128 138
129 139
130 140
131 141
132 142
133 143
134 144
135 145
136 146
137 147
138 148
139 149
140 150
141 151
142 152
143 153
144 154
145 155
146 156
147 157
148 158
149 159
150 160
151 161
152 162
153 163
154 164
155 165
156 166
157 167
158 168
159 169
160 170
161 171
162 172
163 173
164 174
165 175
166 176
167 177
168 178
169 179
170 180
171 181
172 182
173 183
174 184
175 185
176 186
177 187
178 188
179 189
180 190
181 191
182 192
183 193
184 194
185 195
186 196
187 197
188 198
189 199
190 200
191 201
192 202
193 203
194 204
195 205
196 206
197 207
198 208
199 209
200 210
201 211
202 212
203 213
204 214
205 215
206 216
207 217
208 218
209 219
210 220
211 221
212 222
213 223
214 224
215 225
216 226
217 227
218 228
219 229
220 230
221 231
222 232
223 233
224 234
225 235
226 236
227 237
228 238
229 239
230 240
231 241
232 242
233 243
234 244
235 245
236 246
237 247
238 248
239 249
240 250
241 251
242 252
243 253
244 254
245 255
246 256
247 257
248 258
249 259
250 260
251 261
252 262
253 263
254 264
255 265
256 266
257 267
258 268
259 269
260 270
261 271
262 272
263 273
264 274
265 275
266 276
267 277
268 278
269 279
270 280
271 281
272 282
273 283
274 284
275 285
276 286
277 287
278 288
279 289
280 290
281 291
282 292
283 293
284 294
285 295
286 296
287 297
288 298
289 299
290 300
291 301
292 302
293 303
294 304
295 305
296 306
297 307
298 308
299 309
300 310
301 311
302 312
303 313
304 314
305 315
306 316
307 317
308 318
309 319
310 320
311 321
312 322
313 323
314 324
315 325
316 326
317 327
318 328
319 329
320 330
321 331
322 332
323 333
324 334
325 335
326 336
327 337
328 338
329 339
330 340
331 341
332 342
333 343
334 344
335 345
336 346
337 347
338 348
339 349
340 350
341 351
342 352
343 353
344 354
345 355
346 356
347 357
348 358
349 359
350 360
351 361
352 362
353 363
354 364
355 365
356 366
357 367
358 368
359 369
360 370
361 371
362 372
363 373
364 374
365 375
366 376
367 377
368 378
369 379
370 380
371 381
372 382
373 383
374 384
375 385
376 386
377 387
378 388
379 389
380 390
381 391
382 392
383 393
384 394
385 395
386 396
387 397
388 398
389 399
400 400
401 401
402 402
403 403
404 404
405 405
406 406
407 407
408 408
409 409
410 410
411 411
412 412
413 413
414 414
415 415
416 416
417 417
418 418
419 419
420 420
421 421
422 422
423 423
424 424
425 425
426 426
427 427
428 428
429 429
430 430
431 431
432 432
433 433
434 434
435 435
436 436
437 437
438 438
439 439
440 440
441 441
442 442
443 443
444 444
445 445
446 446
447 447
448 448
449 449
450 450
451 451
452 452
453 453
454 454
455 455
456 456
457 457
458 458
459 459
460 460
461 461
462 462
463 463
464 464
465 465
466 466
467 467
468 468
469 469
470 470
471 471
472 472
473 473
474 474
475 475
476 476
477 477
478 478
479 479
480 480
481 481
482 482
483 483
484 484
485 485
486 486
487 487
488 488
489 489
490 490
491 491
492 492
493 493
494 494
495 495
496 496
497 497
498 498
499 499
500 500
501 501
502 502
503 503
504 504
505 505
506 506
507 507
508 508
509 509
510 510
511 511
512 512
513 513
514 514
515 515
516 516
517 517
518 518
519 519
520 520
521 521
522 522
523 523
524 524
525 525
526 526
527 527
528 528
529 529
530 530
531 531
532 532
533 533
534 534
535 535
536 536
537 537
538 538
539 539
540 540
541 541
542 542
543 543
544 544
545 545
546 546
547 547
548 548
549 549
550 550
551 551
552 552
553 553
554 554
555 555
556 556
557 557
558 558
559 559
560 560
561 561
562 562
563 563
564 564
565 565
566 566
567 567
568 568
569 569
570 570
571 571
572 572
573 573
574 574
575 575
576 576
577 577
578 578
579 579
580 580
581 581
582 582
583 583
584 584
585 585
586 586
587 587
588 588
589 589
590 590
591 591
592 592
593 593
594 594
595 595
596 596
597 597
598 598
599 599
600 600
601 601
602 602
603 603
604 604
605 605
606 606
607 607
608 608
609 609
610 610
611 611
612 612
613 613
614 614
615 615
616 616
617 617
618 618
619 619
620 620
621 621
622 622
623 623
624 624
625 625
626 626
627 627
628 628
629 629
630 630
631 631
632 632
633 633
634 634
635 635
636 636
637 637
638 638
639 639
640 640
641 641
642 642
643 643
644 644
645 645
646 646
647 647
648 648
649 649
650 650
651 651
652 652
653 653
654 654
655 655
656 656
657 657
658 658
659 659
660 660
661 661
662 662
663 663
664 664
665 665
666 666
667 667
668 668
669 669
670 670
671 671
672 672
673 673
674 674
675 675
676 676
677 677
678 678
679 679
680 680
681 681
682 682
683 683
684 684
685 685
686 686
687 687
688 688
689 689
690 690
691 691
692 692
693 693
694 694
695 695
696 696
697 697
698 698
699 699
700 700
701 701
702 702
703 703
704 704
705 705
706 706
707 707
708 708
709 709
710 710
711 711
712 712
713 713
714 714
715 715
716 716
717 717
718 718
719 719
720 720
721 721
722 722
723 723
724 724
725 725
726 726
727 727
728 728
729 729
730 730
731 731
732 732
733 733
734 734
735 735
736 736
737 737
738 738
739 739
740 740
741 741
742 742
743 743
744 744
745 745
746 746
747 747
748 748
749 749
750 750
751 751
752 752
753 753
754 754
755 755
756 756
757 757
758 758
759 759
760 760
761 761
762 762
763 763
764 764
765 765
766 766
767 767
768 768
769 769
770 770
771 771
772 772
773 773
774 774
775 775
776 776
777 777
778 778
779 779
780 780
781 781
782 782
783 783
784 784
785 785
786 786
787 787
788 788
789 789
790 790
791 791
792 792
793 793
794 794
795 795
796 796
797 797
798 798
799 799
800 800
801 801
802 802
803 803
804 804
805 805
806 806
807 807
808 808
809 809
810 810
811 811
812 812
813 813
814 814
815 815
816 816
817 817
818 818
819 819
820 820
821 821
822 822
823 823
824 824
825 825
826 826
827 827
828 828
829 829
830 830
831 831
832 832
833 833
834 834
835 835
836 836
837 837
838 838
839 839
840 840
841 841
842 842
843 843
844 844
845 845
846 846
847 847
848 848
849 849
850 850
851 851
852 852
853 853
854 854
855 855
856 856
857 857
858 858
859 859
860 860
861 861
862 862
863 863
864 864
865 865
866 866
867 867
868 868
869 869
870 870
871 871
872 872
873 873
874 874
875 875
876 876
877 877
878 878
879 879
880 880
881 881
882 882
883 883
884 884
885 885
886 886
887 887
888 888
889 889
890 890
891 891
892 892
893 893
894 894
895 895
896 896
897 897
898 898
899 899
900 900
901 901
902 902
903 903
904 904
905 905
906 906
907 907
908 908
909 909
910 910
911 911
912 912
913 913
914 914
915 915
916 916
917 917
918 918
919 919
920 920
921 921
922 922
923 923
924 924
925 925
926 926
927 927
928 928
929 929
930 930
931 931
932 932
933 933
934 934
935 935
936 936
937 937
938 938
939 939
940 940
941 941
942 942
943 943
944 944
945 945
946 946
947 947
948 948
949 949
950 950
951 951
952 952
953 953
954 954
955 955
956 956
957 957
958 958
959 959
960 960
961 961
962 962
963 963
964 964
965 965
966 966
967 967
968 968
969 969
970 970
971 971
972 972
973 973
974 974
975 975
976 976
977 977
978 978
979 979
980 980
981 981
982 982
983 983
984 984
985 985
986 986
987 987
988 988
989 989
990 990
991 991
992 992
993 993
994 994
995 995
996 996
997 997
998 998
999 999
1000 1000

```


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