Fast campus × upstage

Upstage AI Lab

Machine Learning Project | 2024. 12. 06(금)

www.fastcampus.co.kr
Copyright © FAST CAMPUS Corp. All Rights Reserved. 무단전재 및 재배포 금지

목차

01. 팀 소개
팀원 소개 / 협업 방식

02. 프로젝트 개요
목표 수립 / 기술 스택 & 아키텍처 설계

03. 프로젝트 수행 절차 및 방법
데이터셋 및 데이터 처리 / 모델 개발 / 모델 배포 / MLOps 워크 플로우 / 모니터링

04. 회고
결과 / 인사이트 도출 / 향후 계획 / 느낀점

Copyright © FAST CAMPUS Corp. All Rights Reserved. 무단전재 및 재배포 금지

01

팀 소개

팀장/팀원 소개
협업 방식

www.fastcampus.co.kr
Copyright © FAST CAMPUS Corp. All Rights Reserved. 무단전재 및 재배포 금지

* Team 사전오기 : 네번 쓰러져도 다섯번째 다시 일어날 것이다.

팀장

조성지

관심 분야 : 추천 시스템
전공 : 경영학과

역할 : MLFlow 환경설정 및
데이터 크롤링, 프론트엔드 제작

팀원

조혜인

관심 분야 : 데이터 엔지니어링
전공 : 컴퓨터공학

역할 : 모델링 및 서빙

팀원

안서인

관심 분야 : NLP, 의료 도메인
전공 : 컴퓨터공학

역할 : 모델링 및 평가지표 제작

팀원

김태환

관심 분야 : 에듀테크
전공 : 물리학

역할 : 모델링 및 서빙

Copyright © FAST CAMPUS Corp. All Rights Reserved. 무단전재 및 재배포 금지

프로젝트 협업 방식

: MLOps Project

협업 마인드셋 :

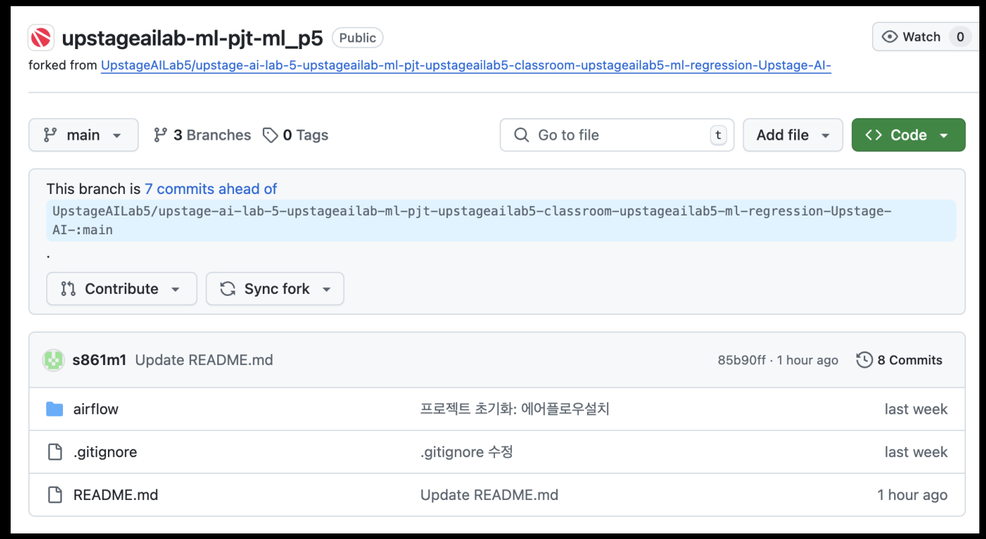
1) 1) 사소한 의견이라도 적극적으로 제시하기
2) 2) 새로 발견한 점이 있으면 공유하기
3) 3) 각자 직접 해보고 어려웠던 점 공유하기
4) 4) (경진대회 이후 추가) 잦은 미팅으로 인한 피로감 개선.
보다 장기적인 프로세스에 도전할 수 있도록 조 모임 횟수 감소, 슬랙 보고 증가.

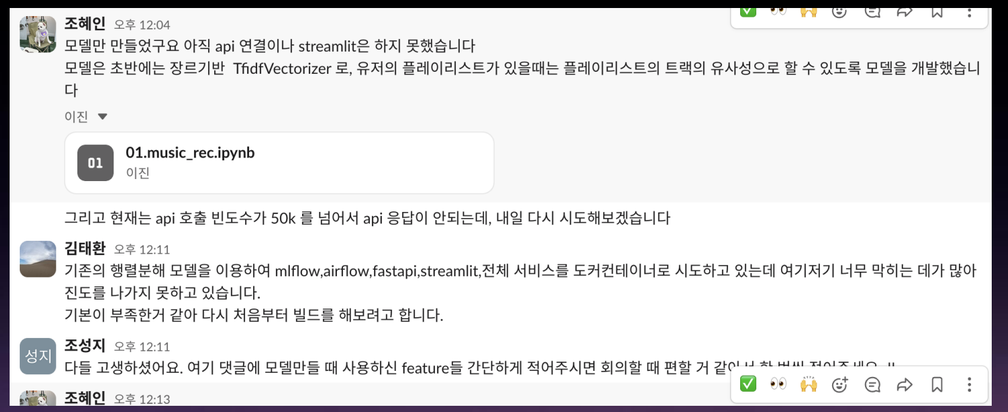
협업 진행 횟수 및 일정 : 진행사항 슬랙으로 공유, 1~2일에 한 번씩 소회의실에서 진행사항 보고

협업 진행 시 생긴 문제점 : 추천 시스템 분야 자체가 생소해서 모델 구축에 어려움이 있었음.
Mlops 환경 세팅 과정에서 문제 발생.

문제 해결 방법 : 추천 시스템에 대한 사전 공부 기간을 둠. 다양한 방식의 추천시스템을 시도한 뒤에 취합

기타 : 환경 세팅 오류 문제는 멘토링 활용.





www.fastcampus.co.kr
Copyright © FAST CAMPUS Corp. All Rights Reserved. 무단전재 및 재배포 금지

02

프로젝트 개요

목표 수립
기술 스택 / 아키텍처 설계

www.fastcampus.co.kr
Copyright © FAST CAMPUS Corp. All Rights Reserved. 무단전재 및 재배포 금지

프로젝트 목표 수립

: MLOps Project | 목표 및 주요 작업

주제

MLOps 프로젝트 | 데이터 전처리부터 모델 서빙까지의 경험

스포티파이 음악 스트리밍 추천 시스템

목표

목표

MLops 플랫폼 사용에 익숙해지기

주요 작업

- Spotify api 크롤링
- Top-k 추천 시스템 구축
- MIflow를 통한 모니터링, FastAPI 및 Streamlit을 통한 시각화

개요

소개 및 배경 설명

사용자 음악 스트리밍 데이터를 기반으로 개인화된 음악 추천 모델을 구축하고 실시간으로 추천을 제공. Batch Serving으로 데이터를 주기적으로 분석하여 추천 결과 업데이트.

기간

2024. 11. 25 ~ 2024. 12.06

www.fastcampus.co.kr
Copyright © FAST CAMPUS Corp. All Rights Reserved. 무단전재 및 재배포 금지

프로젝트 개요

: MLOps Project | 기술 스택 및 아키텍처 설계

기술 스택 요약

프론트엔드

Streamlit

모델 서빙

FastAPI

스토리지

PostgreSQL

실험 추적

mlflow

오케스트레이션

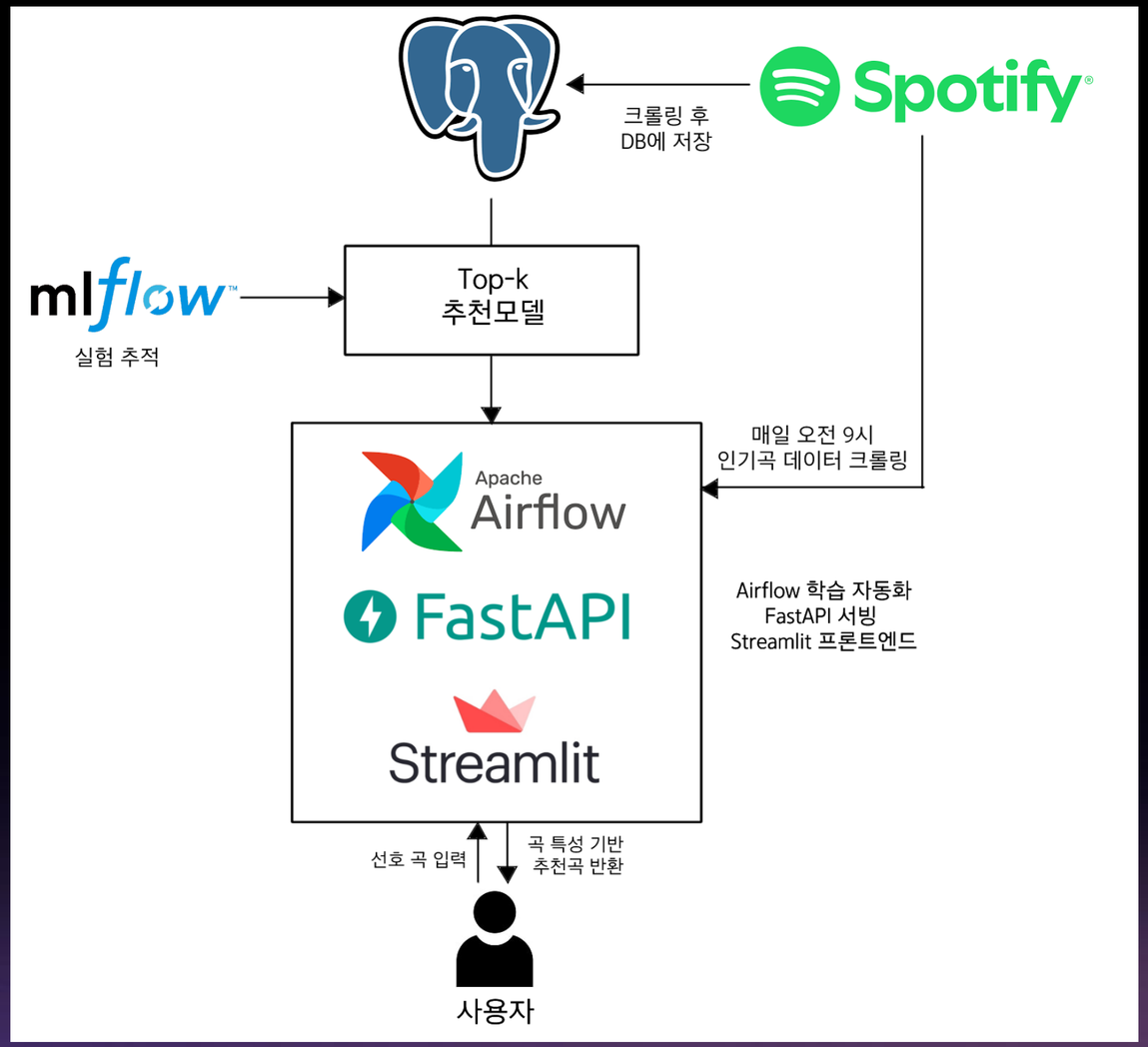
Apache Airflow

www.fastcampus.co.kr
Copyright © FAST CAMPUS Corp. All Rights Reserved. 무단전재 및 재배포 금지

프로젝트 개요

: MLOps Project | 기술 스택 및 아키텍처 설계

- PostgreSQL
  스포티파이 데이터 크롤링 후 저장
- MLFlow
  실험 추적 및 성능지표 관리
- Airflow
  데이터 크롤링 및 실험 자동화
- FastAPI
  모델 서빙
- Streamlit
  프론트엔드 배포

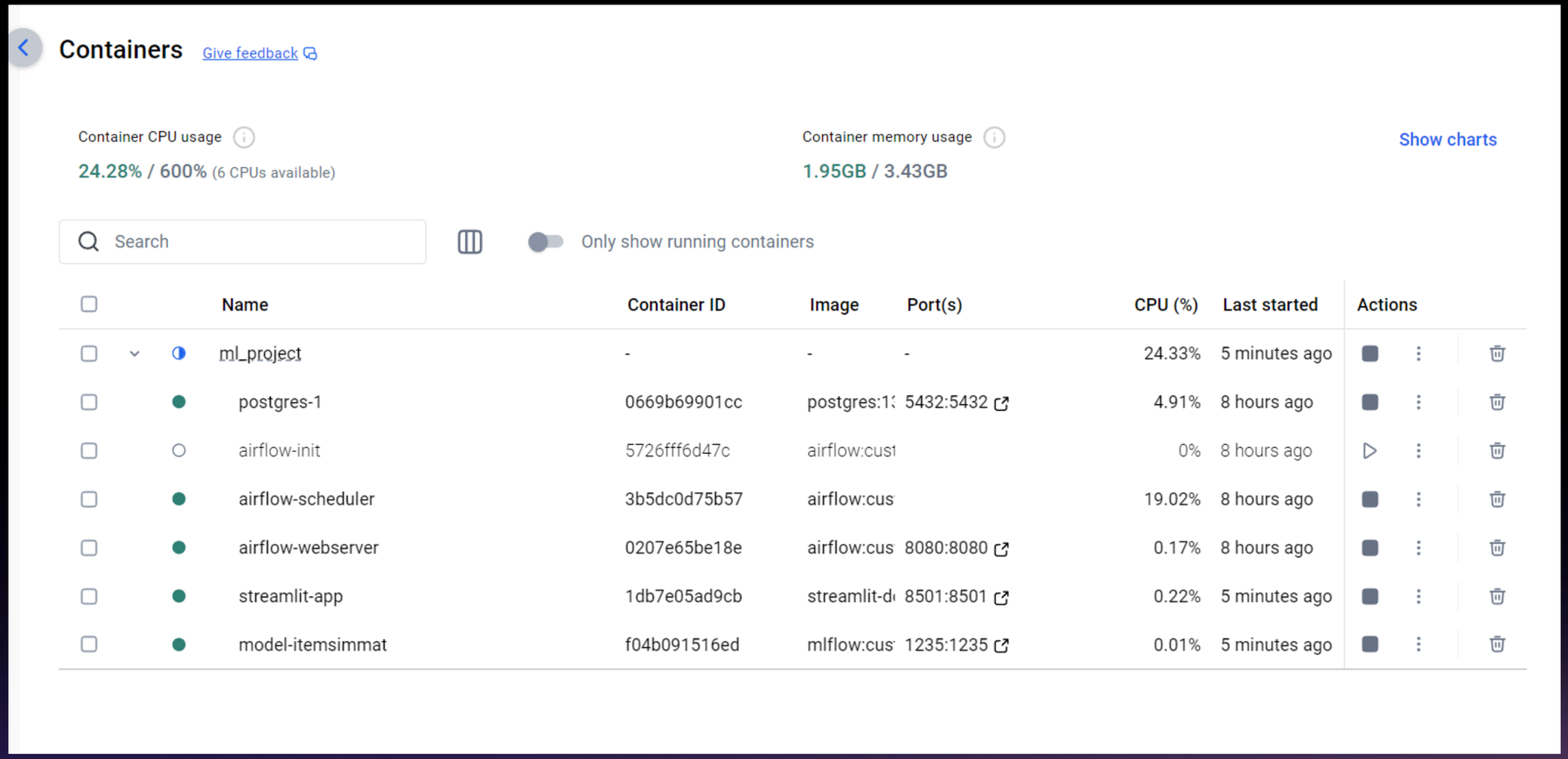


www.fastcampus.co.kr
Copyright © FAST CAMPUS Corp. All Rights Reserved. 무단전재 및 재배포 금지

프로젝트 개요

: MLOps Project | 기술 스택 및 아키텍처 설계

컨테이너 설계



www.fastcampus.co.kr
Copyright © FAST CAMPUS Corp. All Rights Reserved. 무단전재 및 재배포 금지

03

프로젝트 수행 절차 및 방법

데이터셋 및 데이터 처리 / 모델 개발 / 모델 배포 /
MLOps 워크 플로우 / 모니터링

www.fastcampus.co.kr
Copyright © FAST CAMPUS Corp. All Rights Reserved. 무단전재 및 재배포 금지

프로젝트 수행 절차 및 방법

: MLOps Project | 데이터셋 및 데이터 처리

데이터셋 :

1) Spotify 크롤링 데이터

출처 : Spotify Get Audio features API

목표 : 곡의 음악적 특성 기반 곡 추천 시스템 제작

www.fastcampus.co.kr
Copyright © FAST CAMPUS Corp. All Rights Reserved. 무단전재 및 재배포 금지

Upstage AI Lab

프로젝트 수행 절차 및 방법

: MLOps Project | 데이터셋 및 데이터 처리

EDA

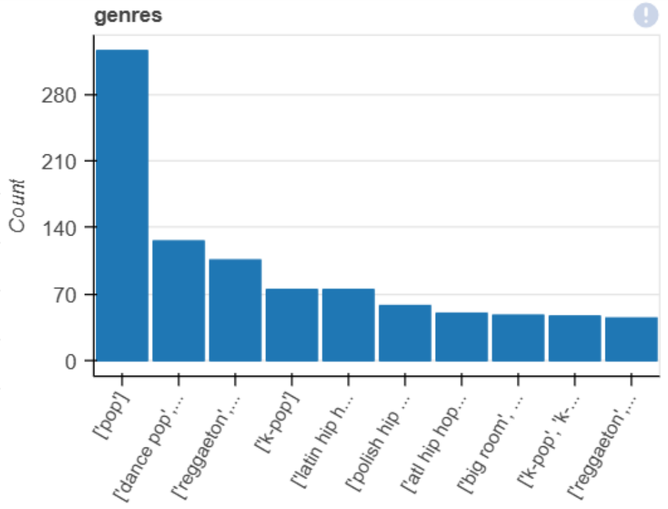
1) Dataprep EDA 자동화 툴 사용

Dataprep report

- 변수 분포 확인 및 피어슨 상관계수 확인
- 사용할 모델(거리 유사도 기반)에 합당한 전처리 작업(Scaling 등)

genres
categorical
Show Details

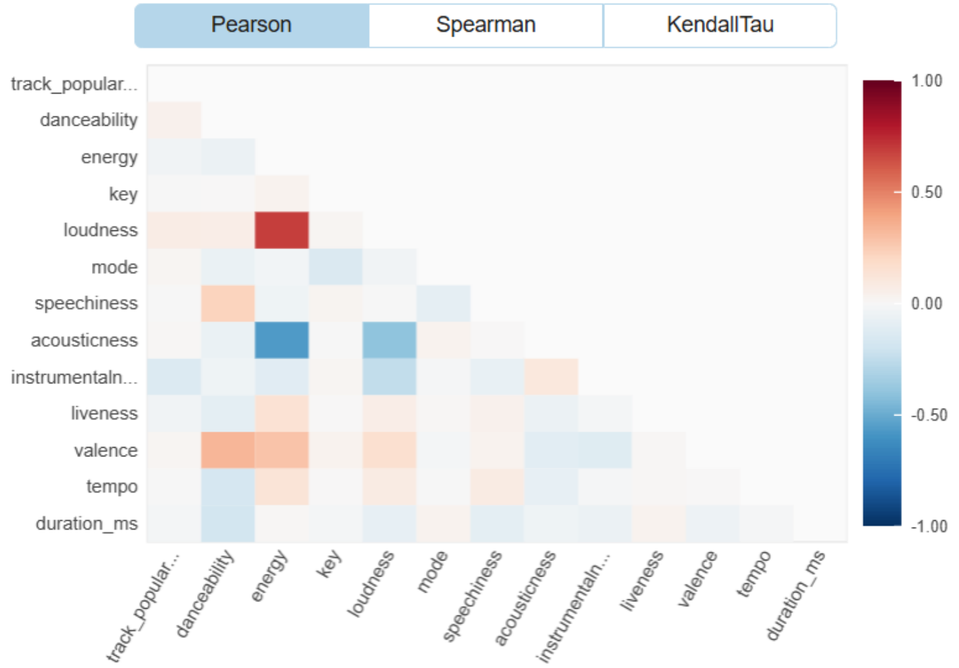
Approximate Distinct Count	2520
Approximate Unique (%)	25.8%
Missing	208
Missing (%)	2.1%
Memory Size	1122318



Overview

Dataset Statistics

Number of Variables	22
Number of Rows	9990
Missing Cells	208
Missing Cells (%)	0.1%
Duplicate Rows	0
Duplicate Rows (%)	0.0%
Total Size in Memory	8.4 MB
Average Row Size in Memory	877.2 B
Variable Types	Categorical: 10 Numerical: 12



www.fastcampus.co.kr
Copyright © FAST CAMPUS Corp. All Rights Reserved. 무단전재 및 재배포 금지

프로젝트 수행 절차 및 방법

: MLOps Project | 데이터셋 및 데이터 처리

데이터 특징

1) Spotify 크롤링, 곡의 음악적 특성을 나타내는 데이터
2) 9990개의 행, 22개의 변수

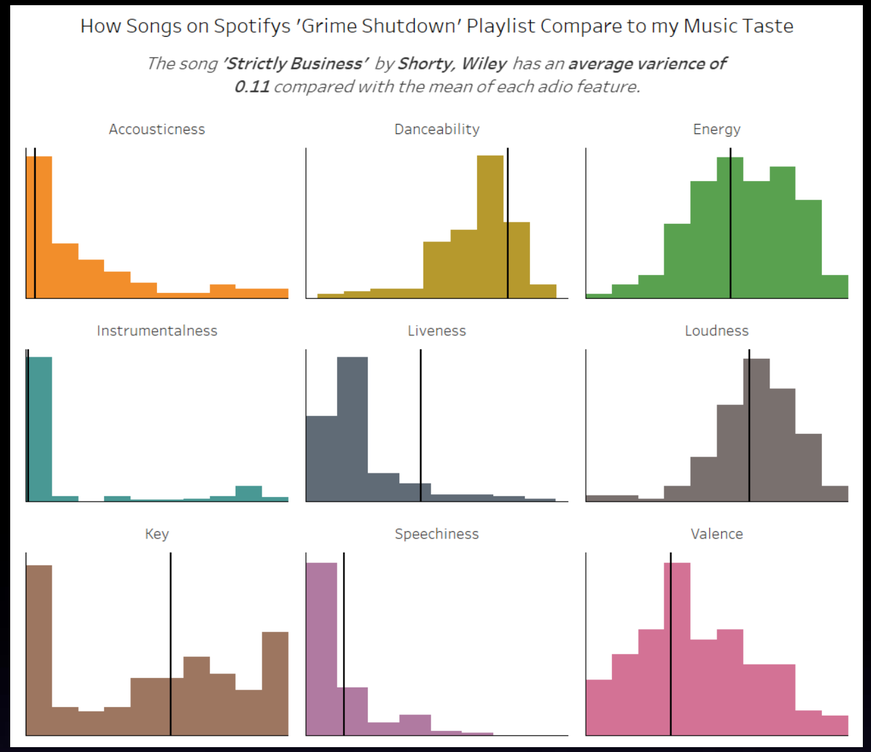
주요 컬럼

track_id : 곡 ID
track_name : 곡 제목
artist : 가수명
genres : 곡 장르 (텍스트)

곡의 음악적 특성 관련 컬럼

- danceability : 춤 추기에 적합한가? (0~1)
- energy : 얼마나 활동적인가? (0~1)
- acousticness : 어쿠스틱 악기 기반인가? (0~1)
- key : 조성 (0~11까지. 0 = C, 1 = C♯/D♭, 2 = D)
- liveness : 라이브에서 연주되었는가? (0~1)
- loudness : 트랙의 전반적인 음량 (dB 기준)

… 포함 13개




```
df.columns
```

✓ 0.0s

```
Index(['index', 'acousticness', 'danceability', 'duration_ms', 'energy',
       'instrumentalness', 'key', 'liveness', 'loudness', 'mode',
       'speechiness', 'tempo', 'time_signature', 'valence', 'target',
       'song_title', 'artist', 'genres'],
      dtype='object')
```


www.fastcampus.co.kr
Copyright © FAST CAMPUS Corp. All Rights Reserved. 무단전재 및 재배포 금지

프로젝트 수행 절차 및 방법

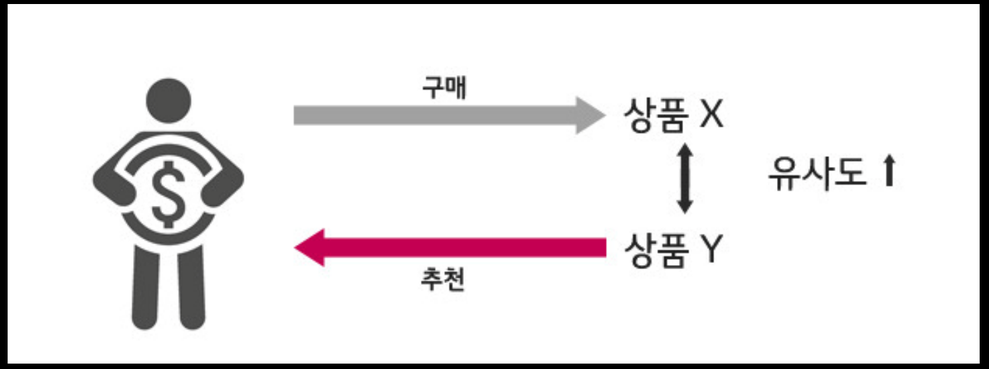
: MLOps Project | 데이터셋 및 데이터 처리

평가 지표 :

1) 코사인 유사도(추천 지표), Human evaluation(모델 성능 지표)

현업 추천시스템의 평가지표 : 매출, 조회수 증감, 유저 선호도 평가 등.
유저층이 계속 변하기 때문에 정량적 점수의 의미가 적고,
모델 배포 이전까지 평가가 어려움.

-> 프로젝트이기 때문에 임시 평가지표 도입
유사도가 높은 아이템을 추천한다는 기본 개념으로부터 착안



https://www.lgcns.com/blog/cns-tech/ai-data/15526/

```
Input = 예뻤어

추천 곡 리스트:
1. Viva La Vida by ['Coldplay'] (유사도: 0.99) -> 5점
2. Alone Again by ['Dokken'] (유사도: 0.99) -> 4점
3. High On Life (feat. Bonn) by ['Martin Garrix', 'Bonn'] (유사도: 0.99) -> 4점
4. Bridges by ['BROODS'] (유사도: 0.99) -> 4점
5. Stay Schemin by ['Rick Ross', 'French Montana', 'Drake'] (유사도: 0.98) -> 2점

평균 3.8점
```


Copyright © FAST CAMPUS Corp. All Rights Reserved. 무단전재 및 재배포 금지

프로젝트 수행 절차 및 방법

: MLOps Project | 데이터 전처리

장르 정보 벡터화 (TF-IDF 가중치 부여)

스트리밍 사이트 가입 시 선호 장르 정보를 물어보는 점에서 착안,
장르(범주형) 데이터 TF-IDF 처리

```python
# 1. 특정 음악을 플레이할때, 그 특정 음악에 대한 추천
def get_recommendations_genre_similarity(df: pd.DataFrame, select_track_name, count):
    vectorizer = TfidfVectorizer()
    tfidf_matrix = vectorizer.fit_transform(df['genres'])

    recommendations = []
    for track_name in select_track_name:
        song_idx = df[df['track_name'] == track_name].index[0]

        genre_sim = cosine_similarity(tfidf_matrix[song_idx], tfidf_matrix).flatten()

        similar_indices = np.argsort(-genre_sim)[1:count+1]
        temp = df.iloc[similar_indices][['track_name', 'artist', 'genres']]

        # Filter out songs that are already in the select_track_name list
        temp = temp[~temp['track_name'].isin(select_track_name)]

        # Check if temp is not empty and matches the length of genre_sim
        if not temp.empty:
            # Ensure that the number of genre similarities matches the number of rows in temp
            temp['genre_similarity'] = genre_sim[similar_indices][:len(temp)]
            recommendations.append(temp)

    return recommendations
```


www.fastcampus.co.kr
Copyright © FAST CAMPUS Corp. All Rights Reserved. 무단전재 및 재배포 금지

프로젝트 수행 절차 및 방법

: MLOps Project | 모델 개발

사용한 모델

Top-K 추천 시스템

사용자가 들었거나 선택한 곡과 DB 내 다른 곡 사이 음악적 특성 기반 cosine 유사도 계산 후 top-k (3, 5, 7 …) 개의 음악을 추천해주는 모델

모델 선택 이유

특징	Top-K	Matrix Factorization
입력 데이터	아이템(곡) 특성 정보	사용자-아이템 상호작용 데이터
출력 데이터	추천 곡	
추천 방식	곡 간 유사도 기반 아이템-아이템 행렬	사용자와 곡 간 선호도 기반 유저-아이템 행렬
콜드 스타트	사용자 없이 곡만으로 추천 가능	새로운 사용자나 아이템 처리가 어려움.
적합한 상황	곡의 특성 데이터가 잘 설계된 경우	사용자-아이템 평가 데이터가 충분한 경우

- 주어진 데이터 내에서 학습시킬 수 있는 모델.
- 콜드스타트 이슈에서 자유로움
- 알고리즘이 가볍고 간단함.

www.fastcampus.co.kr
Copyright © FAST CAMPUS Corp. All Rights Reserved. 무단전재 및 재배포 금지

프로젝트 수행 절차 및 방법

: MLOps Project | 모델 개발

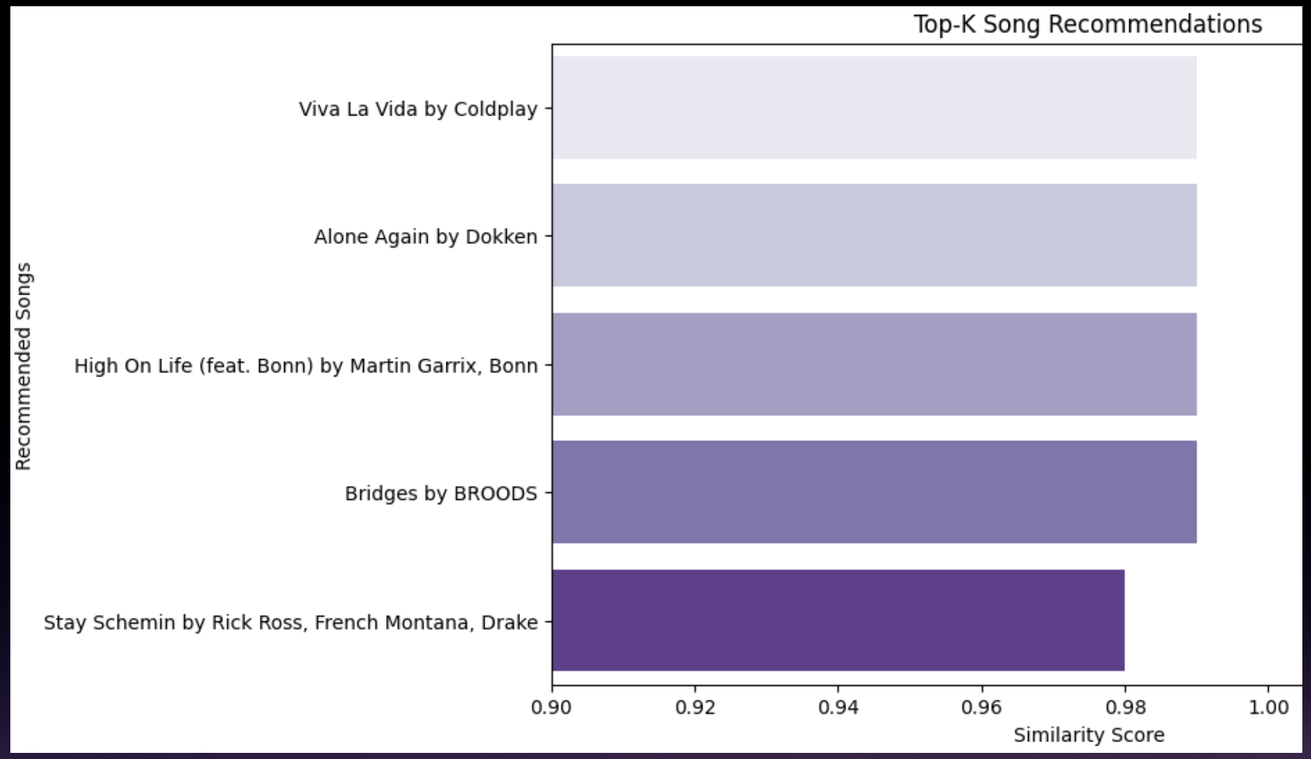
모델 훈련 및 성능

훈련 과정
- 데이터 준비: 데이터 전처리, TF-IDF 가중치 부여

훈련
- 아이템-아이템 행렬 기반 Top-K 모델 학습

모델 성능 지표
Cosine 유사도

이후 Pkl 라이브러리를 통해 모델 저장



www.fastcampus.co.kr
Copyright © FAST CAMPUS Corp. All Rights Reserved. 무단전재 및 재배포 금지

프로젝트 수행 절차 및 방법

: MLOps Project | 모델 배포

배포 방법

사용한 모델:
- FastAPI (배포)
- Streamlit (프론트엔드)

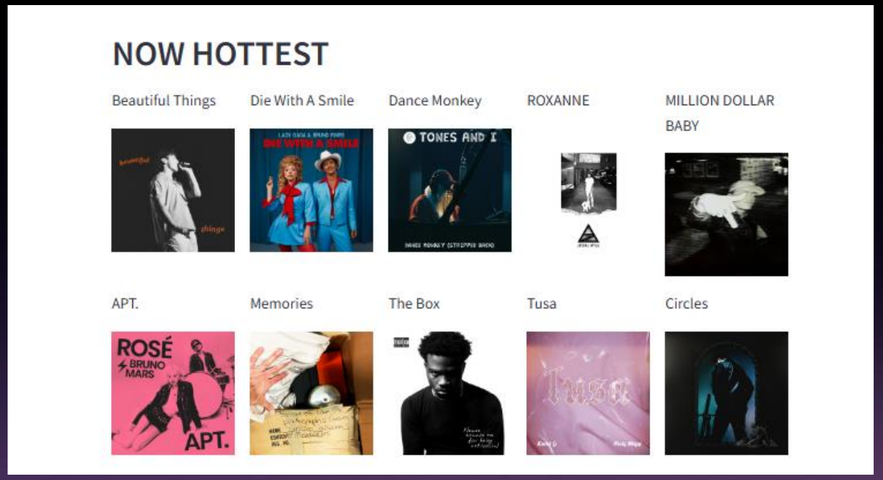
배포 과정 :
훈련된 모델 FastAPI로 배포 및 Streamlit 서빙.
최근에 재생한 곡 혹은 (콜드스타트) 특정
곡으로부터
곡 특성 코사인 유사도 기반 top-k 음악 반환

배치 서빙

배치 서빙을 위한 Airflow 사용

- 데이터 크롤링 :
  매일 오전 9시, 인기곡 top 10 데이터
- 모델 업데이트 :
  인기곡 데이터 사용, 모델 추가 학습

```python
# 추천 곡을 생성하는 함수
def get_recommendations(user_id):
    try:
        # Spotify API를 통해 사용자의 트랙 가져오기
        listened_tracks = get_user_tracks()

        # 매핑된 트랙 필터링
        mapped_tracks = [track_to_index[track] for track in listened_tracks if track in track_to_index]

        # 매핑된 트랙이 없을 경우 처리
        if not mapped_tracks:
            return {"error": "No valid tracks found in the user's listened tracks."}

        # 사용자 벡터 생성
        temp_user_interactions = csr_matrix(
            (np.ones(len(mapped_tracks)),
             ([0] * len(mapped_tracks), mapped_tracks)),
            shape=(1, len(track_id_unique))
        )
        temp_user_vector = svd.transform(temp_user_interactions)[0]

        # 추천 곡 생성
        recommended_track_ids = recommend_songs_exclude_listened(temp_user_vector, listened_tracks, n_recommendations=5)
        recommended_track_names = convert_track_id_to_name(recommended_track_ids)

        return {"tracks": recommended_track_names}
    except Exception as e:
        return {"error": f"An error occurred: {str(e)}"}
```



www.fastcampus.co.kr
Copyright © FAST CAMPUS Corp. All Rights Reserved. 무단전재 및 재배포 금지

프로젝트 수행 절차 및 방법

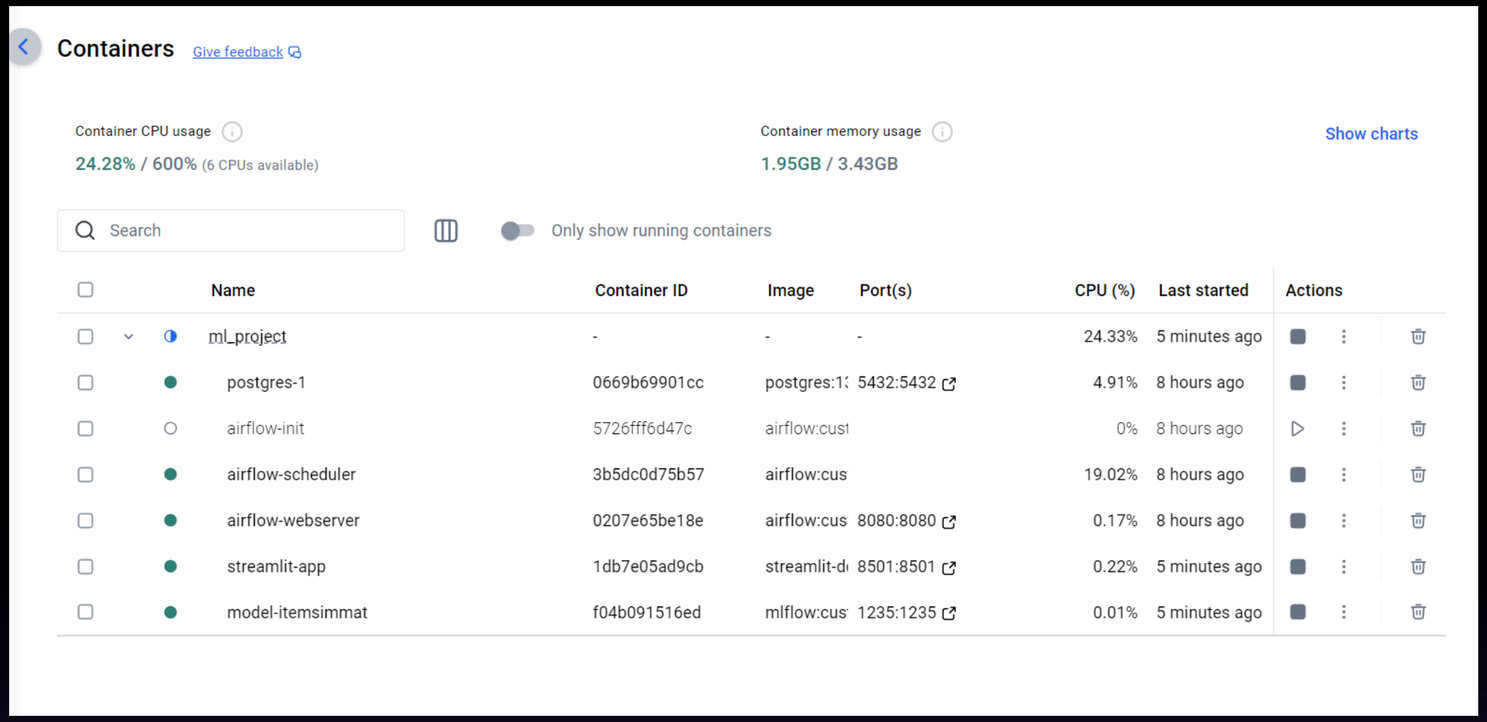
: MLOps Project | MLOps 워크 플로우

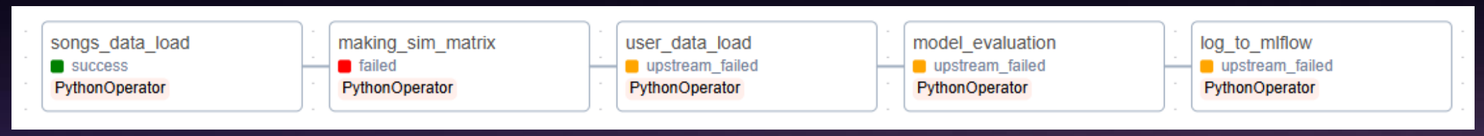
CI/CD 파이프라인

Airflow 사용

CI 단계 : 코드 통합
- airflow-scheduler, airflow-webserver workflow (스케줄링 및 자동화)
- ML 파이프라인과 데이터 준비
- Postgres (데이터 저장)

CD 단계 - 모델 배포:
- streamlit-app (프론트엔드 제공)
- model-itemsimmat (배포된 모델 실행, API 요청 처리)





www.fastcampus.co.kr
Copyright © FAST CAMPUS Corp. All Rights Reserved. 무단전재 및 재배포 금지

프로젝트 수행 절차 및 방법

: MLOps Project | 모니터링

모니터링

성능 개선 변화 추이

MLflow 사용

실시간 트래킹
- Autolog 활성화
- 실험 로그와 성능 지표를 실시간으로 기록하고 분석.
- 다양한 파라미터와 측정값을 비교하여 모델의 성능을 평가.

- 주요 지표 : Cosine 유사도

```python
# MLflow로 모든 하이퍼파라미터 및 모델 비교
def evaluate_models_with_hyperparameters():
    # 데이터 로드 및 분리
    data = load_data()
    train_data, test_data = split_data(data)

    # 하이퍼파라미터 목록
    hyperparams = [3, 5, 7]

    # 모델 설정
    models = {
        "CosineSimilarity": train_and_evaluate_cosine,
        "EuclideanDistance": train_and_evaluate_euclidean,
    }

    # MLflow 자동 로깅 활성화
    mlflow.autolog()
```

			Metrics	
Run Name	Created	Duration	avg_test_distanc	avg_test_similari
EuclideanDistance_param_7	12 minutes ago	5.6s	0.032506488...	-
EuclideanDistance_param_5	12 minutes ago	5.8s	0.029376214...	-
EuclideanDistance_param_3	12 minutes ago	11.4s	0.025326303...	-
CosineSimilarity_param_7	12 minutes ago	1.0s	-	0.934684925...
CosineSimilarity_param_5	12 minutes ago	1.1s	-	0.923239160...
CosineSimilarity_param_3	12 minutes ago	1.0s	-	0.942523776...

Copyright © FAST CAMPUS Corp. All Rights Reserved. 무단전재 및 재배포 금지

04

회고

결과 / 인사이트 도출 / 향후 계획
느낀점

www.fastcampus.co.kr
Copyright © FAST CAMPUS Corp. All Rights Reserved. 무단전재 및 재배포 금지

프로젝트 회고

: MLOps Project | 결과 및 향후 계획

1
* 최종 결과

Mlops 관점:

다양한 Mlops 툴 사용,
도커 컨테이너 환경 세분화

2
* 도전 과제 및 해결

다양한 mlflow 환경 구축을 시도했으나,
(airflow 자동화, mlflow 업데이트 등)
Dags 복잡도 상승, 충분하지 않은 시간.

->모든 툴이 원활하게 돌아가지 않음.
디버깅과의 싸움.

3
* 인사이트 도출

프로젝트의 핵심인 모델 개발에 집중

-> 많은 아쉬움이 남는다.

4
* 개선 방향

프로젝트 이후 추가 디버깅.
다음 프로젝트에 Mlops 적극적으로 활용

www.fastcampus.co.kr
Copyright © FAST CAMPUS Corp. All Rights Reserved. 무단전재 및 재배포 금지

프로젝트 회고

: MLOps Project | 결과 및 향후 계획

1
* 최종 결과

모델 관점:

Top-k 아이템 기반 곡 추천 모델

주어진 데이터 특성을 살린
아이템-아이템 유사도 기반

2
* 도전 과제 및 해결

추천 시스템 평가 지표 도입의 어려움
현업에서 사용되는 지표는 클릭 수, 매출.
프로젝트 단위의 평가지표가 필요함.

3
* 인사이트 도출

Human evaluation 평가지표 도입

4
* 개선 방향

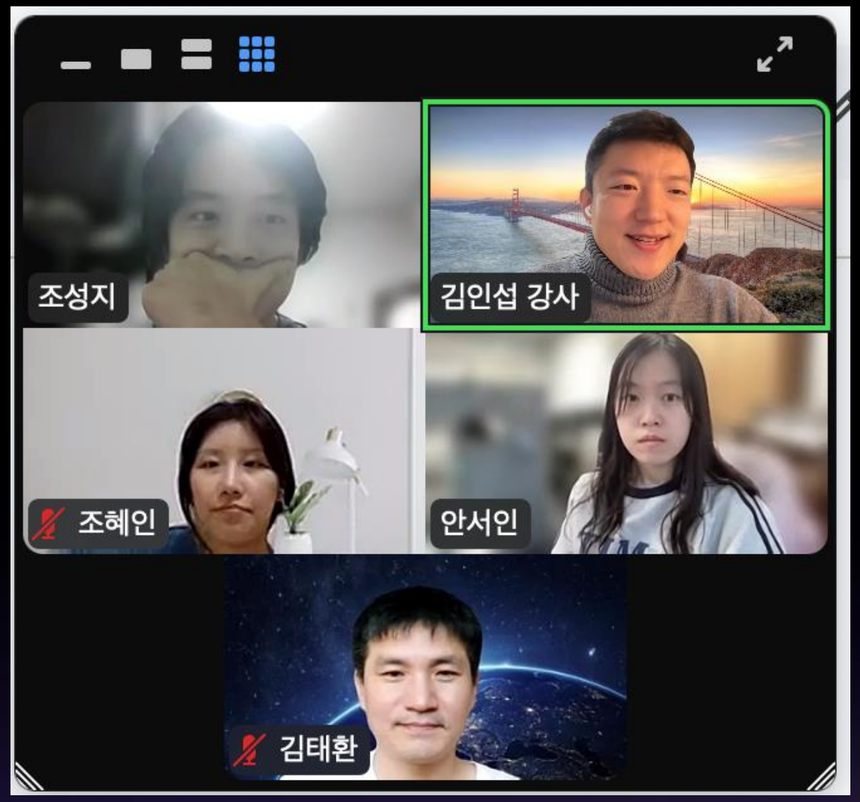
보다 객관적인 평가지표 사용
사용자-아이템 행렬 기반 추천 시스템
고도화

www.fastcampus.co.kr
Copyright © FAST CAMPUS Corp. All Rights Reserved. 무단전재 및 재배포 금지

프로젝트 진행 소감

: MLOps Project | 느낀점

느낀점



*** 조성지** 매일 발생하는 버그와의 싸움이었지만 개인적으로 정말 많은 것을 배울 수 있었던 기간이었다.
막연히 어렵게 느껴졌던 mlop 워크플로우 구축이 조금 익숙하게 다가오는 계기가 될 것 같다.

*** 조혜인** MLOPS 너무 어려웠지만 시간이 지날수록 뭘 하고 있는지 내가 어떻게 실무에 적용할 수 있을 지 배울 수 있었다 .
다음번에는 추천 시스템이 아니라 결과를 도출해야하는 프로젝트라면 MLOPS를 통해 많은 도움을 받을 수 있을 것 같아 유익한 시간이였다

*** 안서인** Mlops 환경 세팅은 코드만으로 해결되는 문제가 아니라서 많은 어려움이 있었습니다.
직접 적용해보지 못한 부분이 많아서 아쉬웠습니다. 다음 프로젝트에는 이번에
적용하지 못했던 mlops 툴을 성공적으로 다룰 수 있으면 좋겠습니다.

*** 김태환** 이번 프로젝트를 하면서 서비스 구조를 구현하는 게 많이 힘들었습니다.
컴퓨터 지식과 코딩이 많이 부족하다는 것을 뼈져리게 느꼈습니다.
앞으로 더 많은 과정이 남아 있는데 부족한 기초를 더 다듬으며 가야겠다는 각오입니다.

Copyright © FAST CAMPUS Corp. All Rights Reserved. 무단전재 및 재배포 금지

Life-Changing Education

감사합니다.

Copyright © FAST CAMPUS Corp. All Rights Reserved. 무단전재 및 재배포 금지