PORTFOLID DATA ANALYSIS _ LEESUYEON

노션:https://www.notion.so/suyeun/83900228acec4c9ab1dd5f438bd22421?v=f47f21ede5694d9b96dd154fb6b66cd6

깃허브:https://github.com/LEESUSUSUSU

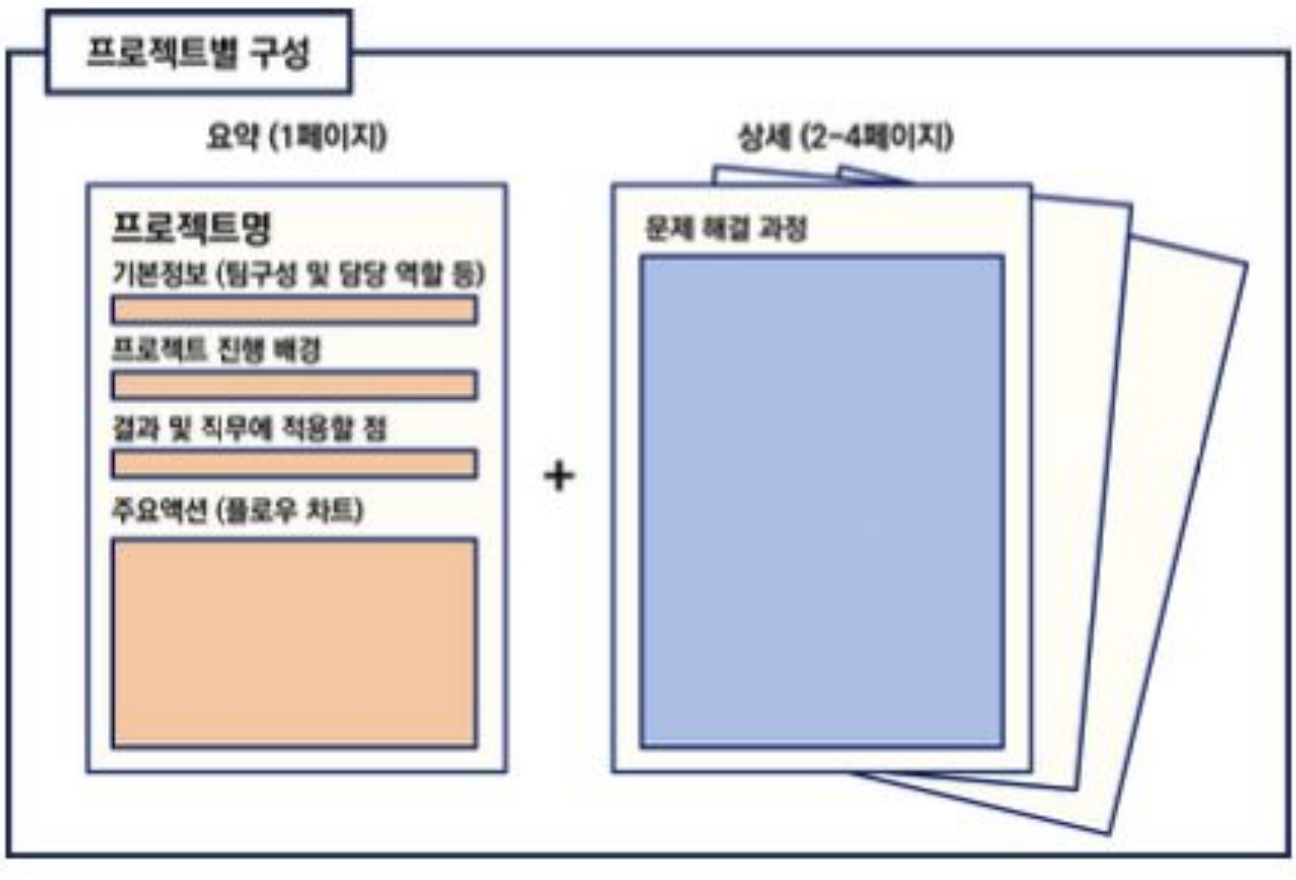
포트폴리오

본 포트폴리오의 구성

각 프로젝트를 중요 내용을 소개한 요약 페이지와 전체 내용을 담은 상세 페이지로 구성했습니다. 상세 페이지에는 단계별로 GitHub or Notion 을 첨부하여 코드를 확인하실 수 있도록 했습니다.

기차 대차 이상소음 데이터의 대해서는 보안상 올릴 수 없음을 이해 부탁드립니다.

또한, 그간의 역량 성장 과정을 보실 수 있도록 수행한 순서대로 프로젝트를 배치하였습니다.



목차

본 포트폴리오는 총 4개의 데이터 분석 프로젝트를 포함하고 있습니다.

대회

데이터 경진대회 (인공지능 기반 기차 대차 이상 소음 위치 감지 시스템 구축)

- **기간**: 2023년 9월 ~ 2023년 11월
- **역할**: 데이터 분석, 모델링 , 알고리즘
- **기술 스택: Python, NumPy, Pandas, Scikit-Learn, PyTorch, Tensorflow, Librosa**
- **성과**: 최종 발표 단계까지 진행 함으로써 여러 모델링의 기본적 지식을 습득하고 FFT알고리즘에 대해서 이해함 , 데이터들이 시사하는 정보가 무엇인지 파악하는 것이 중요하다는 알게됨
- **설명**: 타겟이 없는 상태에서 유의미한 인사이트를 발견

프로젝트

유퀴즈 데이터를 이용한 머신러닝 분석

- **기간**: 2024.02월 ~ 2024.02 월
- **역할**: 날짜 별 EDA , 시청률 모델링
- **성과**: 시청률 분석에 정확도를 예측할 수 있는 모델링
- **기술 스택**: Python, Pandas, Scikit-Learn
- **설명**: 유퀴즈 라는 프로그램을 통해 우리가 아는 사람이 나온다면 시청률 과 조회수가 어떻게 될까?

시각 장애 보조를 위한 딥러닝 기반 장애물 인식

- **기간**: 2024.03월 ~ 2024.04월 3일
- **역할**: 위험감지
- **성과**: yolo 모델의 이해
- **기술 스택**: Python, Pandas,yolo 8
- **설명**: 시각장애인들을 위한 보행 보조 프로그램

신용카드 사기 탐지 모델

- **기간**: 2024.04 ~ 2024.05
- **역할**: 신용카드 사기 탐지 모델 , 분석
- **성과**: 불균형 데이터의 이해
- **기술 스택**: Python, Pandas,Scikit-Learn,TensorFlow,Imbalanced-learn
- **설명**: 신용카드 사기를 탐지하여 국민의 안전한 신용카드 사용을 위함

기차 대차 이상 소음 위치 감지 시스템 구축

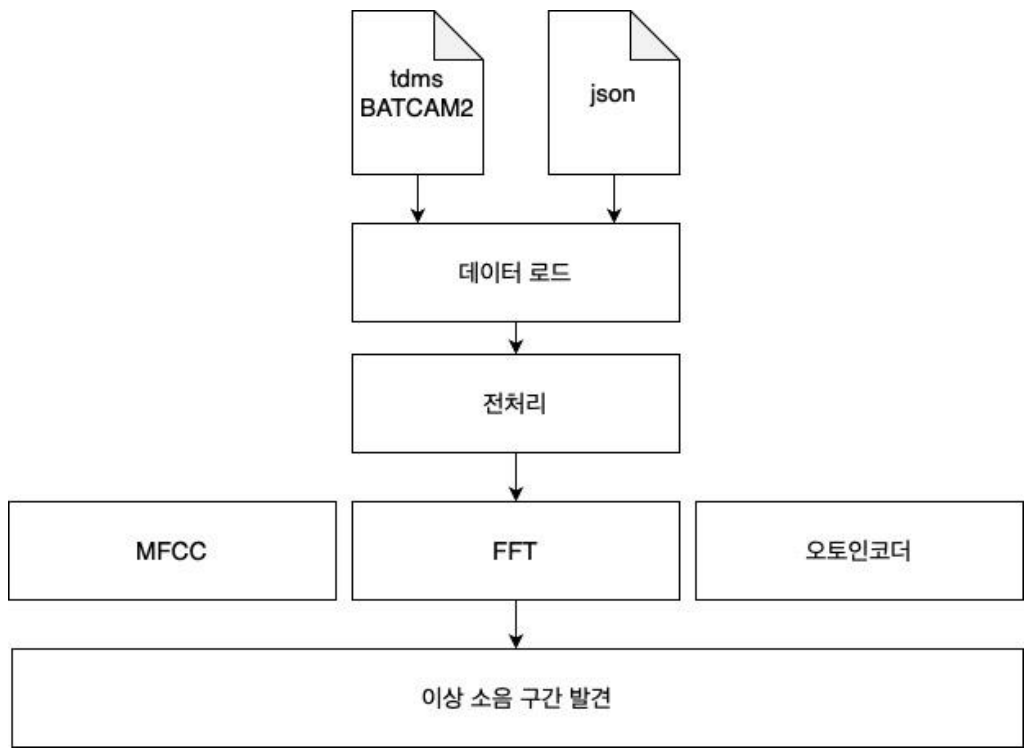
요약

- 담당 역할
  - 전처리
  - **fft** 알고리즘을 이용한 모델링
  - 기차대차 이상판별 위치 알고리즘

프로젝트 진행배경

- 대회명: 2023 DATA·AI 분석 경진대회
- 주관: 한국과학기술정보연구원
- 목적: 국민의 생활에서 발생하는 다양한 비언어적 소리를 데이터로 축적하여 공공 교통안전 현안 문제를 해결하기 위해 소리데이터 융합·활용하여. 사회문제 해결
- 요구사항: 기존에 수집된 기차의. 소음 데이터를 활용, 소음 데이터 분석을 위한 AI 알고리즘 설계, 감지된 이상 소음의 위치 및 위험도 표시 기능

프로세스



채널별 음향 데이터

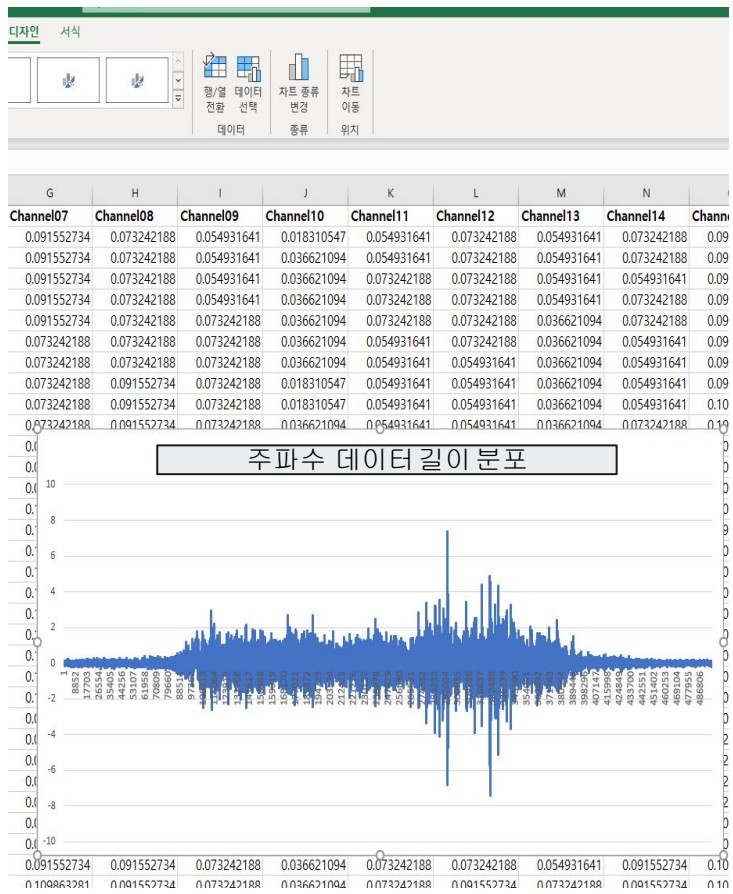
데이터 구조 분석

수집된 데이터:

- 7일 동안 수집된 215개의 TDMS 파일과 82개의 JSON 파일

TDMS 파일:

- 각 TDMS 파일은 다수의 채널(channel)로 구성
- 각 TDMS 파일의 row 길이는 파일마다 다름
  - row의 길이를 통해 영상의 시간을 확인 가능
  - (이는 기차 속도 계산에 필요)



G	H	I	J	K	L	M	N
Channel07	Channel08	Channel09	Channel10	Channel11	Channel12	Channel13	Channel14
0.091552734	0.073242188	0.054931641	0.018310547	0.054931641	0.073242188	0.054931641	0.073242188
0.091552734	0.073242188	0.054931641	0.036621094	0.054931641	0.073242188	0.054931641	0.073242188
0.091552734	0.073242188	0.054931641	0.036621094	0.073242188	0.073242188	0.054931641	0.054931641
0.091552734	0.073242188	0.054931641	0.036621094	0.073242188	0.073242188	0.054931641	0.073242188
0.091552734	0.073242188	0.073242188	0.036621094	0.073242188	0.073242188	0.036621094	0.073242188
0.073242188	0.073242188	0.073242188	0.036621094	0.054931641	0.073242188	0.036621094	0.054931641
0.073242188	0.073242188	0.073242188	0.036621094	0.054931641	0.054931641	0.036621094	0.054931641
0.073242188	0.091552734	0.073242188	0.018310547	0.054931641	0.054931641	0.036621094	0.054931641
0.073242188	0.091552734	0.073242188	0.018310547	0.054931641	0.054931641	0.036621094	0.054931641
0.073242188	0.091552734	0.073242188	0.036621094	0.054931641	0.054931641	0.036621094	0.073242188
0.091552734	0.091552734	0.073242188	0.036621094	0.073242188	0.073242188	0.054931641	0.091552734
0.109863281	0.091552734	0.073242188	0.036621094	0.073242188	0.091552734	0.073242188	0.091552734

메타 데이터

포지션 정보 :

- 0m 지점에서 BatCam2 주파수 카메라가 설치된 위치 23m
- 기차가 센서에 진입한 후(0m) 기차 촬영 시작, 33.2m 지점에서 촬영 종료

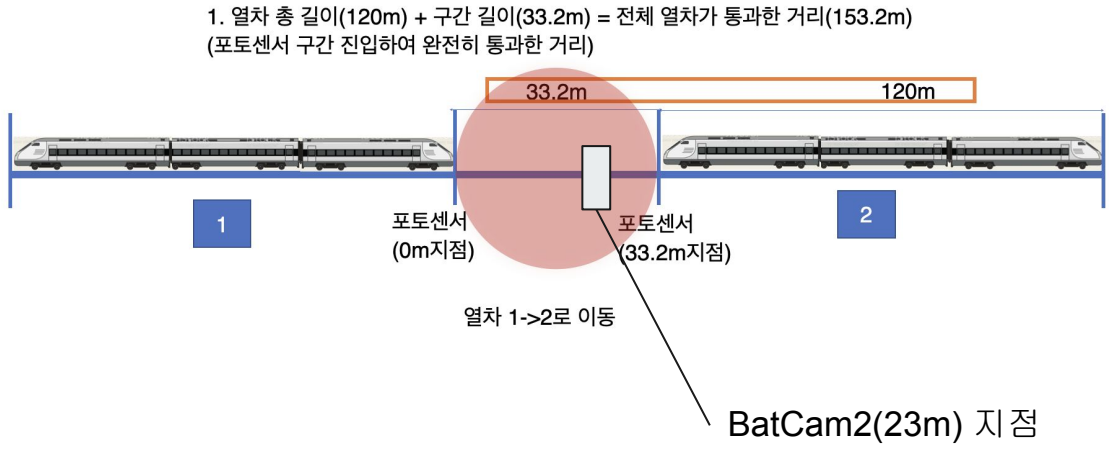
기차 속도 계산:

- 거리를 이용하여 기차 속도 계산
  - 0m 지점에서 센서 진입 시점 기록
  - 33.2m 지점에서 촬영 종료 시점 기록
  - 두 시점 간의 시간을 측정하여 기차의 속도를 계산


```
{
    "title_s206": "test_01.tdms",
    "title_batcam2": "test_01.tdms",
    "Creator": "SM Ins.",
    "Year": "2022",
    "Date": "1102",
    "Train": "수소열차",
        "Length": 44,
        "Car_num": 2,
    "Horn": "Yes",
        "Position": 22,
    "Place": "오송시험선로",
    "Photo_sensor_positions": [0, 33.2],
    "S206_position": 22,
    "Batcam_position": 23
}
```


horn 정보:

- 경적소리의 유무



데이터 전처리

1-1 길이 정리

1-2 pca를 통한 대표채널 선정

데이터의 무거움

- **문제**: 대량의 데이터로 인해 데이터 무거움
- **해결**: 데이터의 효율적인 처리를 위해 데이터셋을 최적화하고 카메라에 잡히지 않는 기차 대차의 소리는 필요가 없기때문에 제거
- **채널 및 파일 별 row 길이 상이**
- **문제**: 각 채널과 파일마다 row의 길이가 다름
- **해결**: 가장 높은 피크값을 기준으로 앞으로 10초, 뒤로 10초의 길이로 맞추어 row의 길이를 통일함. (기차가 카메라에 다다르면 가장 높은 주파수를 가짐 - 모든 파일 공통사항)

```
시작 인덱스: 3644197
------
잘라낸 데이터: [0.02270508 0.31030273 0.36706543 ... 1.0822754  0.71899414 0.77197266]
------
Channel04의 데이터:
Channel04의 데이터 길이: 512000
```

채널마다 받아들이는 주파수 인지 불가

- **문제**: 각 채널마다 받아들이는 주파수 마스킹 되어있는 데이터
- **해결**: 공통되지 않고 문제가 되는 채널을 1개를 추출하여 분석 (pca 기법 활용)
- 데이터의 양이 많아 모델을 돌리는데에 한계가 있었음, 따라서 문제가 있는 데이터를 주로 분석함.

2-1. 배경음 제거를 위한 주파수 도메인 변환

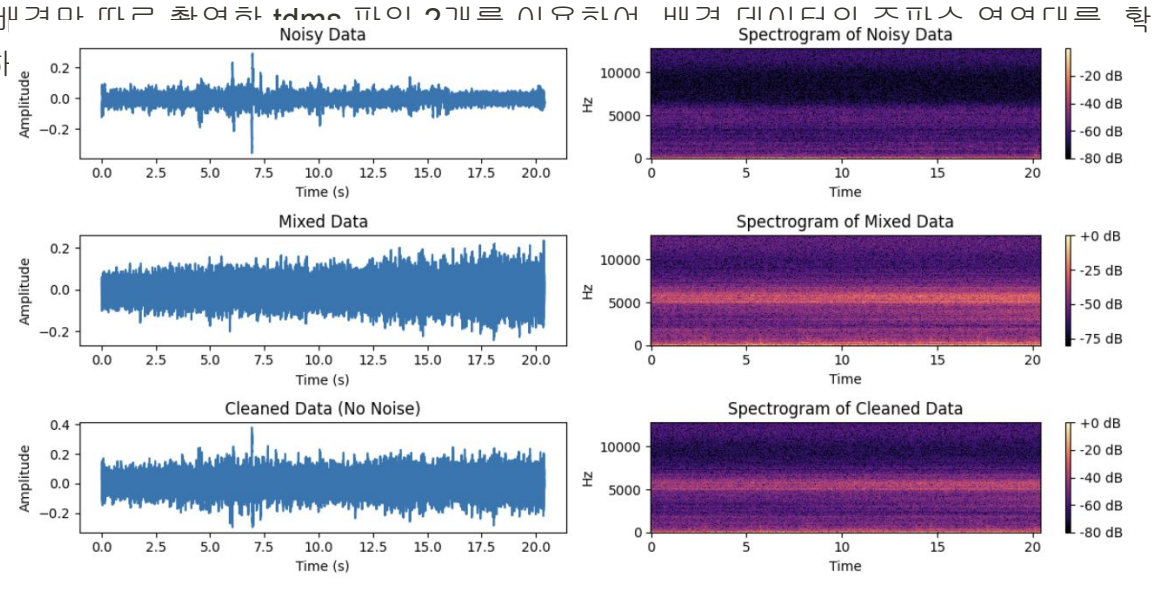
2-2

(-) 연산으로 잡음 제거.

음향데이터의 대표 채널을 선정한 후 대표. 채널의 배경음(bg.tdms) 제거를 위해 FFT(푸리에 고속변환 : 신호를 시간 도메인에서 주파수 도메인으로 전환하는 음향데이터 조정 방식)를 사용하여 주파수 도메인으로 변환.

(-) 연산을 이용하여 음향데이터에서 bg를 제거

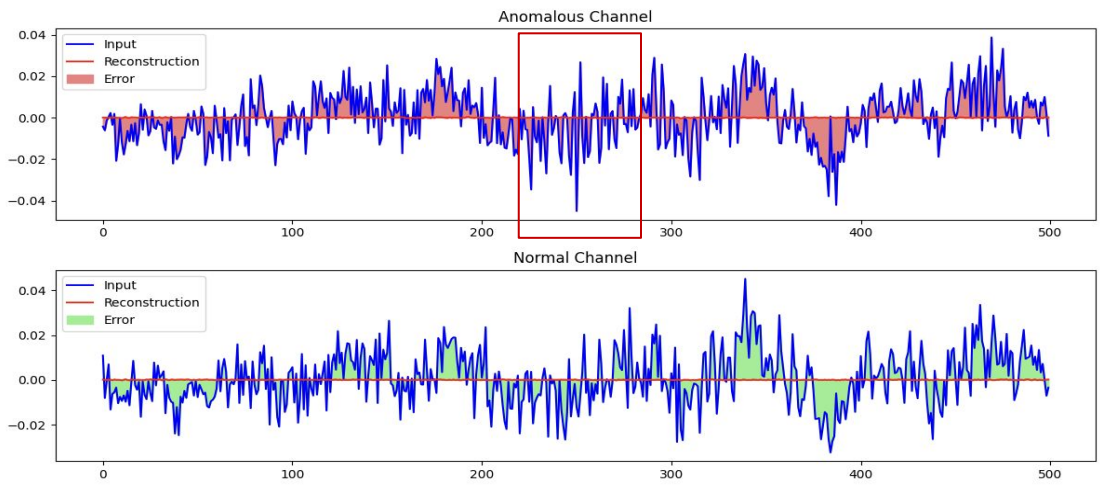
배경만 따로 촬영한 tdms 파일 2개를 이용하여 배경 데이터의 주파수 영역대를 확인하



모델

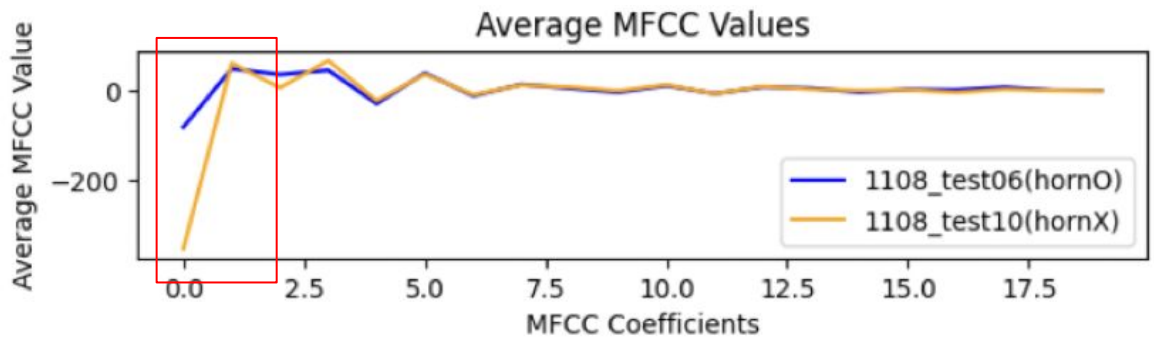
모델 선정	• **목적**: 타겟이 없는 모델의 일관성을 확인하고자 다양한 모델을 구축하여 일반화 성능을 확인하고 이는 horn 이외의 이상 주파수를 찾아 문제 대차를 발견하는 데 중점을 둠 • **방법**: 여러 모델을 비교 및 검증하여 가장 적합한 모델을 선정, 이를 통해 데이터의 특성을 잘 반영하고, 이상 주파수를 효과적으로 탐지할 수 있는 모델을 찾음

오토인코더[**Baseline 모델**]	• **목적**: 주어진 입력 차원에 대해 인코더의 마지막 차원을 1/512로 줄이는 오토인코더 모델을 사용하여 학습
	• **방법**:
	  ○ **재구성 오류 계산**: 모델이 입력 데이터를 재구성하면서 발생하는 오류를 계산하여 이상 감지 임계값을 설정
	  ○ **이상 데이터 탐지**: 설정된 임계값을 기반으로 이상 데이터를 탐지
	• **결론**: 주어진 데이터의 주파수를 분석한 결과, 특정 영역에서 이상 소음이 발생하는 것을 발견. 이를 통해 주파수 도메인에서의 분석이 효과적임을 확인함.

푸리에 변환 및 오토인코더 적용	• **목적**: 주파수 분석을 통해 데이터의 주파수 특성을 파악하고, 오토인코더 모델을 적용
	• **방법**:
	  ○ **푸리에 변환**: 모든 사물은 일정한 주파수를 갖는다. 이를 기반으로 일정한 주파수를 추출
	  ○ **오토인코더 모델 적용**: 추출된 주파수를 이용하여 오토인코더 모델을 적용
	 
	
	임계값 기반 데이터 확인
	• **목적**: 일정 임계값 이상인 데이터를 확인하여 대차를 분석
	• **결론**:
	  ○ 오토인코더 기본 모델을 사용하여 대차를 확인한 결과, 문제가 발생한 m(거리)의 상의 하였으나 문제가 일어난 부분의 대차는 동일 함 (대차 1대당 20m)

결론

horn을 찾는 MFCC 모델 추가적으로 확인

MFCC	• **목적**: 다양한 악기 소리를 분별하는 데 용이한 MFCC(Mel-Frequency Cepstral Coefficients) 모델을 활용하여 horn 소음을 추가적으로 확인 • **방법**: ○ **소음 탐지**: 특정 주파수 영역에서 발생하는 horn 소음 확인  **최종 결론** • **오토인코더와 MFCC 모델 비교**: MFCC 모델과 오토인코더 모델이 각각 다른 문제 영역을 탐지. 이는 각 모델이 다른 주파수 특성을 기반으로 동작하기 때문 • **결론**: horn 이외의 문제를 발견하는 것이 이번 분석의 주된 목적. 오토인코더 모델은 horn 이외의 문제를 탐지하는 데 성공하여, 주파수 도메인에서의 분석이 유효함을 확인,따라서 오토인코더가 horn이 아닌 다른 문제를 찾은 것은 올바른 접근임을 확인했습니다.

결론

- 데이터 대회와 같은 경쟁 환경은 다양한 접근 방식과 시각을 경험할 수 있는 기회를 제공. 이를 통해 자신의 데이터 분석 방법론을 넘어서 보다 폭넓은 관점을 개발하고, 창의적이며 유연한 문제 해결 능력을 키울 수 있었음. 따라서 다양한 시각과 접근 방식을 수용하고 이해하는 것이 데이터 과학자로서 성장하는 데 중요한 시간 이였다고 생각함

느낀점

- **다양한 접근 방식과 시각의 중요성**
  - 데이터 대회를 통해 다른 팀들과의 비교 과정에서 각 팀이 데이터를 바라보는 시각과 접근 방식이 상당히 다르다는 것을 알게 확인함.이를 통해 다양한 해결책을 모색하는 데 큰 도움을 받음

유퀴즈 데이터를 이용한 머신러닝 분석

요약

- 팀구성 및 기여도 : 3명 / 34 %
- 담당 역할
  - 수기 데이터 수집
  - EDA (회차 , 날짜별, 계절별)
  - 시청률 모델링

프로젝트 진행배경

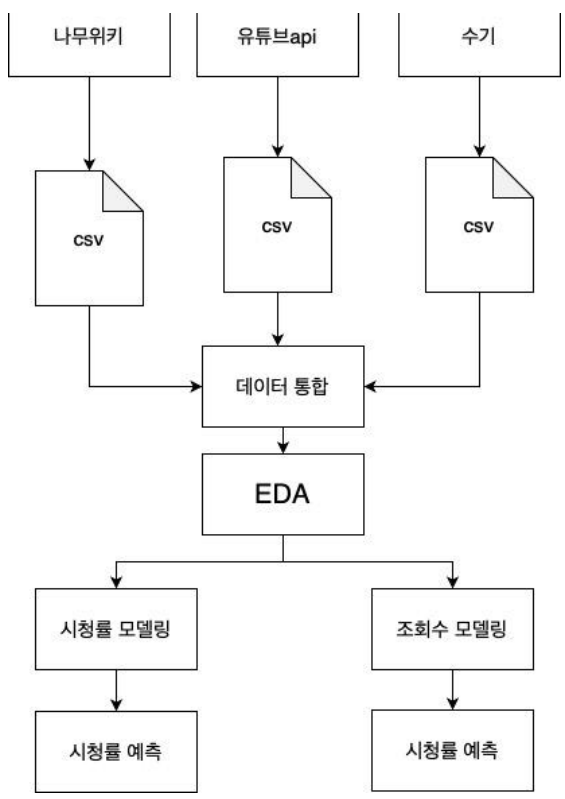
프로젝트를 진행하며 수기로 데이터를 수집, 이를 바탕으로 EDA 후 모델링 까지 만듬으로써 전반적인 데이터 모델링의 흐름을 익히고자 하였음

수기 데이터 수집 :방송 회차별, 날짜별 출연진 및 프로그램 특성 데이터를 수기로 수집
EDA (탐색적 데이터 분석): 회차별, 날짜별 데이터에 대한 탐색적 분석 수행
시청률 모델링: 수집된 데이터를 기반으로 시청률과 조회수를 예측할 수 있는 모델 개발 및 검증

프로젝트 개요

1. 프로젝트명: **유퀴즈 유튜브와 생방송의 조회수와 시청률 예측 모델 만들기**
3. 수행 기간: 3주 (2024.01 ~ 2024.02)
4. 목표: 유퀴즈의 출연진들의 데이터를 넣어서 조회수와 시청률 예측해 보기

프로세스



데이터 수집

1. 나무위키

• 시즌, 회차, 날짜, 주제, 출연자, 시청률, 직업, 성별, 나이, 수상여부, 인지도

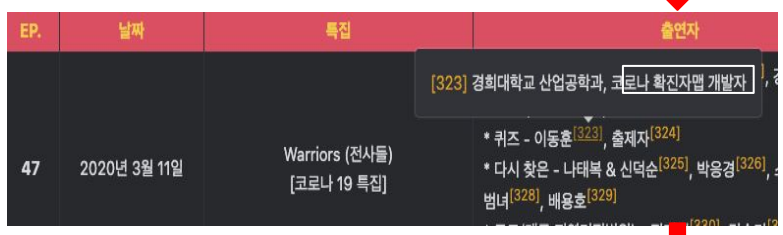
2. 유튜브 api

• 제목, 조회수, 좋아요 수, 댓글 수, 재생시간, 구독자 수 , 회차

3.직접 수집

• 직업군, 성별, 나이,수상여부,인지도

upload_date	subscriber_count	likes	comments	term	job	top
20200729.0	0	6186.0	224.0	1290.0		Tha
20200729.0	0	0.0	0.0	1290.0		Tha
20200729.0	0	1.0	1.0	1290.0		Tha
20200902.0	100000	292.0	25.0	1255.0		Tha

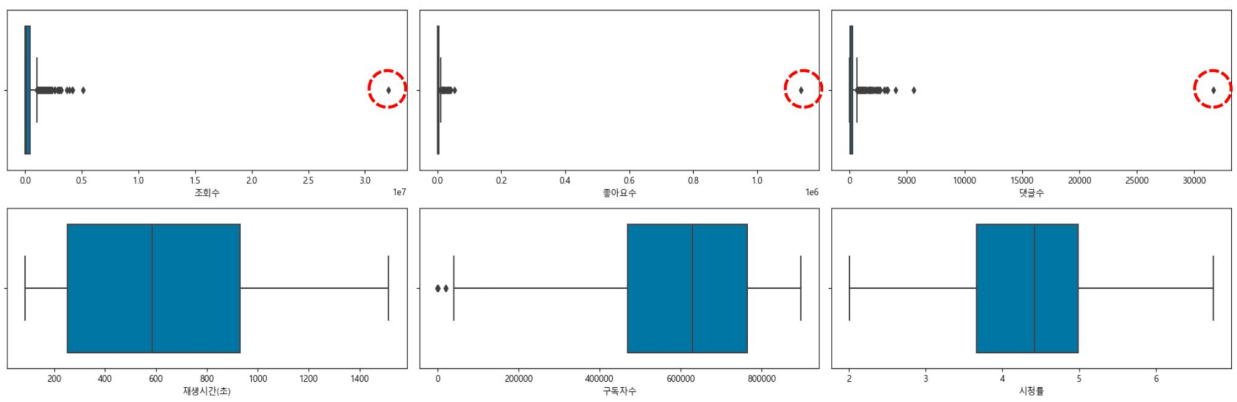


upload_date	subscriber_count	likes	comments	term	job	top
20200729.0	0	6186.0	224.0	1290.0	전문기술	Tha
20200729.0	0	0.0	0.0	1290.0	기타	Tha
20200729.0	0	1.0	1.0	1290.0	기타	Tha
20200902.0	100000	292.0	25.0	1255.0	기타	Tha

EDA

이상치(Outliers) 확인 -BTS

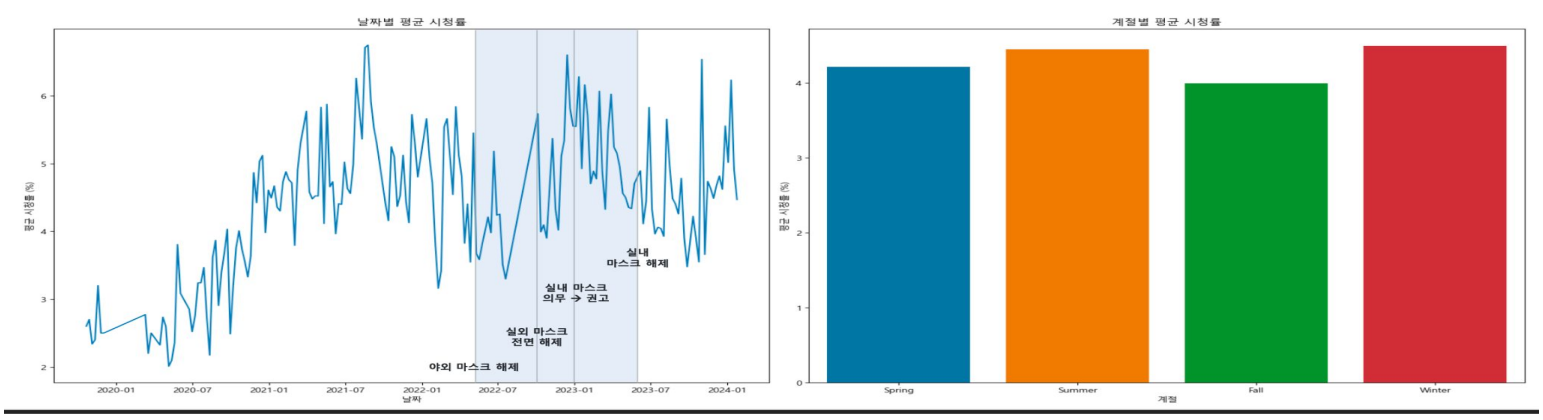
• 이상치 확인
조회수, 좋아요수, 댓글수의 경우 상위 이상치 값이 나타남
이는 다른 데이터들보다 훨씬 많은 조회수, 좋아요수, 댓글수를 가지고 있음을 알 수 있음. (bts)



계절적 변화에 따른 시청률 확인

2023년부터 실내 마스크 착용 의무가 권고 사항으로 변경됨. 코로나19에 대한 경계심이 낮아지면서 사람들의 외부 활동이 증가함. 이 변화는 우상향하던 시청률의 증가세가 둔화되거나 멈추는 원인이 됨.

외부 활동하기 좋은 봄과 가을에 따른 평균 시청률이 소폭 하락하는 추세를 확인함.



시청률 머신러닝

X = df[['직업', '성별', '나이', '구독자수', '수상여부', '인지도']]
y = df[['시청률']]

모델	MAE	MAPE
GradientboostingRegressor	0.28	6.38
XGBRegreesor	0.30	7.17
RandomForestRegressor	0.45	10.39

효과적인 데이터 예측을 위한 시도

1. 날짜 데이터를 사인과 코사인으로 변환하여 새로운 컬럼으로 추가
2. RobustScaler, StandardScaler, MinMaxScaler 스케일링을 시도 하여 최적의 하이퍼파라메터 찾음

X = df[['직업', '성별', '나이', '구독자수', '수상여부', '인지도']]
y = df[['시청률']]

모델	StandarScaler	MinMaxScaler	RobustScaler
GradientboostingRegressor	**0.23 / 0.05%**	**0.23 / 0.05 %**	**0.2 / 0.04 %**
XGBRegreesor	**0.15 / 0.03 %**	**0.17 / 0.04 %**	**0.16 /0.04 %**
RandomForestRegressor	**0.8 /0.2 %**	**0.79/ 0.2 %**	**0.79 / 0.2 %**

프로젝트 완료 후 개별적으로 다시 확인함.최적화를 진행하지 않고 특성만 추가
시간적 특성을 이해하고자 계절(봄, 여름, 가을, 겨울) 특성을 먼저 추가하고, 그 다음에 날짜 데이터를 sin, cos으로 변경함.

모델	계절성 추가 전 MSE	계절성 추가 전 MAE	계절성 추가 후 MSE	계절성 추가 후 MAE	sin,cos 컬럼 수정 후 MSE	sin,cos 컬럼 수정 후 MAE
랜덤 포레스트	0.1373	0.2122	0.1319	0.2030	0.1186	0.1856
그래디언트 부스팅	0.2419	0.3580	0.2413	0.3570	0.2585	0.3550
XGBoost	0.2799	0.3777	0.2760	0.3749	0.2656	0.3620
SVM	0.5312	0.5410	0.5312	0.5410	0.5312	0.5410

조회수 머신러닝

X = df[['직업', '성별', '나이', '구독자수', '수상여부', '인지도']]
y = df[['시청률']]

모델	MAE	MAPE
GradientboostingRegressor	351866.50	2282.52 %
XGBRegreesor	323278.39	3079.86%
RandomForestRegressor	321701.03	3403.23%

결론: 시청률 4.7 조회수 50만 예측

느낀점

- 데이터를 수집하고 어떻게 가공 하느냐에 따라 데이터의 인사이트를 얻을 수 있는 방법이 다양함을 알았음.
- 단순히 모델이 왜 잘 되지 않을까? 라고 생각하지 않고 잘 되지 않는 이유에 대해 고민하며 데이터에 대한 이해도를 높일 수 있었음

시각 장애 보조를 위한 딥러닝 기반 장애물 인식

요약

- 팀구성 및 기여도 : 4명 / 25 %
- 담당 역할
  - 위험탐지 코드 개발
  - 베이스 모델 (오류 수정)
  - 라벨링 작업

프로젝트 진행배경

- 시각장애인들을 돕기위해 딥더링을 이용한 프로그램을 진행하고자 하였음
- 휠체어 이용자들이 보도주행에 있어 위험을 미리 감지하고자 하였음

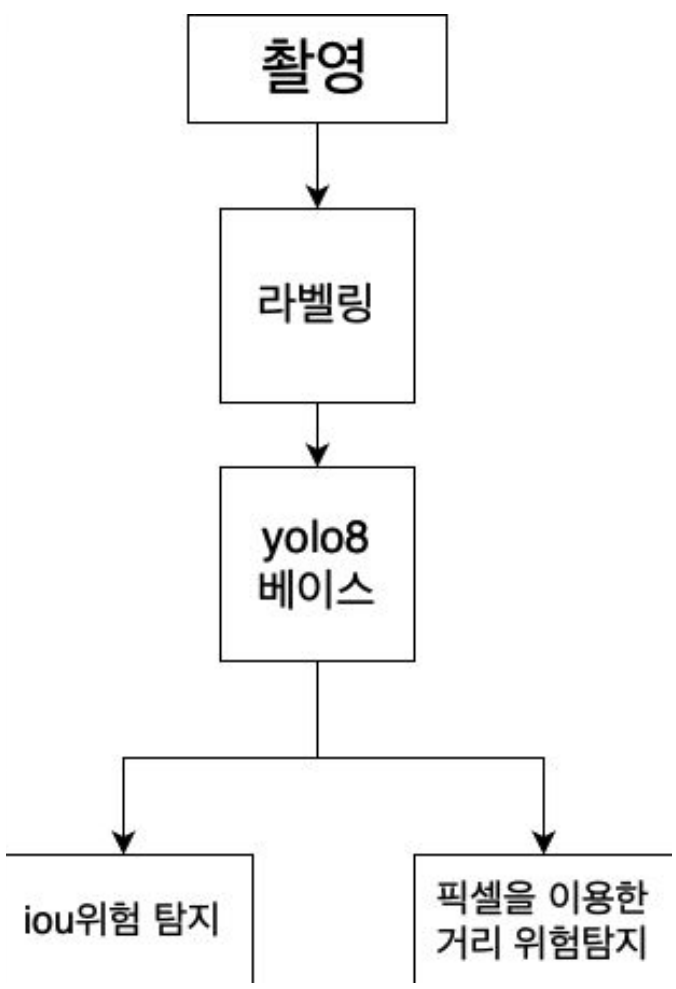
문제 개요

프로젝트명: **시각 장애 보조를 위한 딥러닝 기반 장애물 인식**

수행 기간: 4주 (2024.03 ~ 2024.04)

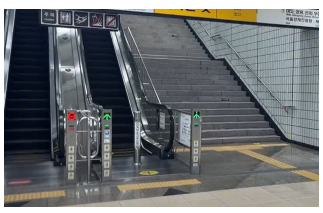
목표: 시각 장애 보조를 위한 위험탐지

프로세스

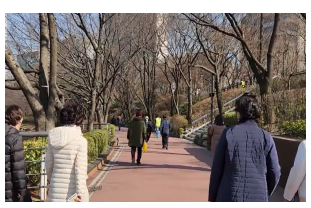


데이터 수집

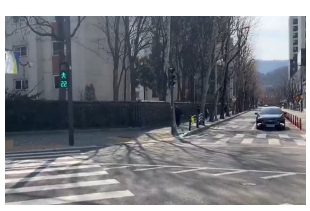
1. 지하철, 경전철



2. 공원, 골목길



3.횡단보도,등 외



데이터 라벨링

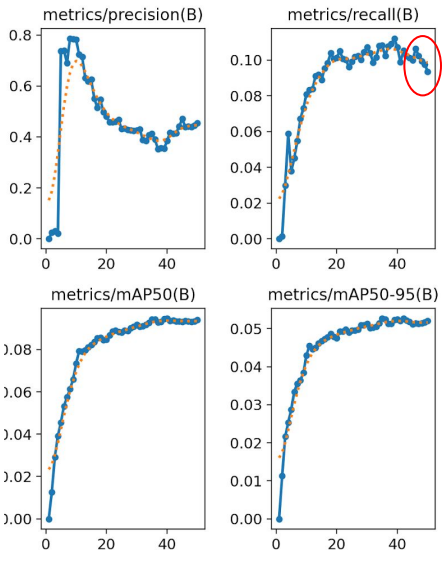
23가지의 Class를 정하고, 수집된 영상 데이터를 3초 단위 프레임으로 나눈 뒤

각 Class 당 최소 1,000개수집을 목표로 라벨링 직접 수행 _ json

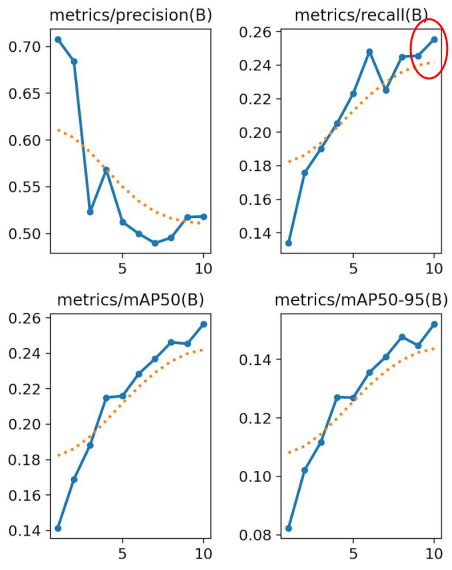
클래스 번호	클래스명		라벨링기법	가이드라인 이슈
0	person	사람	bbox	50% 미만 객체는 잡지 않음
1	car	모든 4륜 구동 차량	bbox	거울에 비친 차량 제외
2	bicycle	자전거	bbox	
3	scooter	전동킥보드	bbox	
4	motorcycle	오토바이	bbox	
5	traffic light	신호등	bbox	불빛이 보이는 경우 6번 선행 후 멀티클래스
6	green	초록불		
7	red	빨간불		
d	yellow	노란불		
9	road	도로/바닥	segmentation	발이 닿는 모든 바닥을 라벨링
10	manhole	맨홀	bbox	열려있을 경우만 라벨링
11	tree	나무	bbox	카메라의 위치에서 도로밖에 있는 경우는 제외 이미지 가로축 50% 이상 넘어간 나무제외
12	dog	개	bbox	
13	cat	고양이	bbox	
14	trash	쓰레기	bbox	쓰레기봉투들 라벨링
15	bus_stop	버스정류장	bbox	정류장표지판을 라벨링
16	train_station	지하철역	bbox	지하철역 표지판을 라벨링
17	utility_pole	전봇대	segmentation	이미지 가로축 50% 이상 넘어간 전봇대 제외
18	bench	벤치 등 모든 의자	bbox	
19	stair	계단	bbox	바닥 부분만 라벨링
20	elevator	엘리베이터	bbox	엘리베이터 문을 라벨링
21	fire hydrant	소화전	bbox	
22	obstacle	그 외 모든 장애물	bbox	

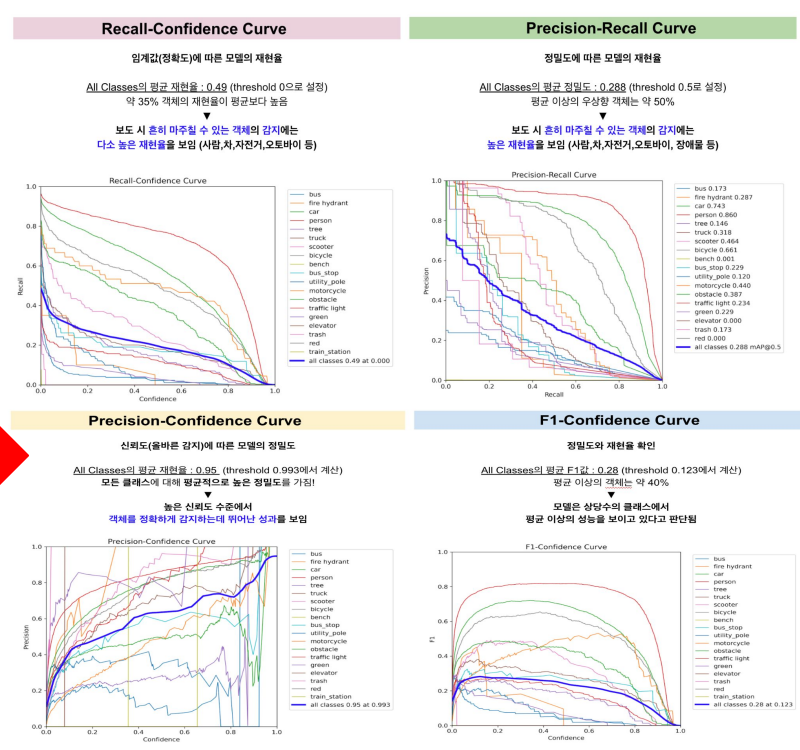
v8 학습 과정 요약

1차class 목록 40여개
100epoch



2차 필요없는 클래스 제거 후 재 학습
10epoch





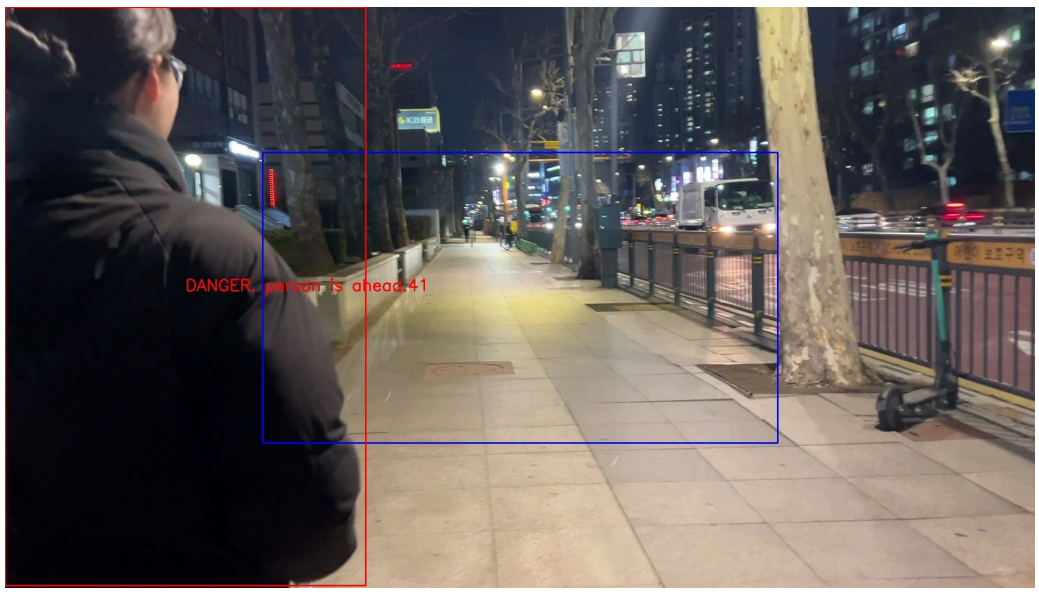
3 차 학습의 컨퓨전 행렬 (Confusion matrix) 확인 시 3차 학습 대비 성능이 **소폭 개선**됨

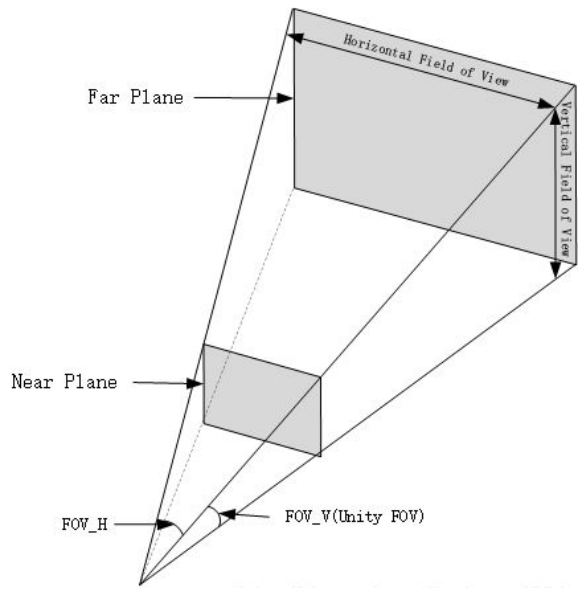
Train	train/box_loss	train/cls_loss	train/dfl_loss	metrics/precision(B)	metrics/recall(B)	metrics/mAP50(B)	metrics/mAP50-95(B)	val/box_loss	val/cls_loss
2차	0.80254	0.68715	0.90292	0.69482	0.65834	0.69639	0.56378	0.72743	0.56906
3차	0.75168	0.48834	0.88301	0.46231	0.2769	0.27593	0.15708	1.196	1.2335

Test 검증

3D 렌더링에서 관찰점을 중앙으로 모으는 것을 "중심 정렬" 또는 "카메라 중심 정렬"이라고 함

관찰자 시점에서 중심 정렬된 네모 박스는 사용자의 **시선을 대변**하여 위험한 물체나 상황을 시각적으로 명확하게 보여줌으로써 대응 능력을 향상시켜 줄 것으로 가정





중심 정렬 네모박스를 기준으로 **약 30% 이상 가까워**지는 경우,

[객체 라벨명]이 포함된 [Danger] 경고 메시지를 표시하여 **위험을 감지할 수 있도록 개발**

결론

- YOLOv8을 활용해 4차 학습 결과 일부 지표에서는 성능이 저하되었지만, mAP50-95에서의 증가와 같이 일부 지표에서는 향상된 결과를 얻음

느낀점

- YOLOv8을 활용한 객체탐지 과정을 경험함 으로써 딥러닝 모델의 구축 및 튜닝에 대한 이해가 높아짐
- 객체탐지 모델의 성능평가 지표 해석과 모델 개선 방안 도출에 대해 경험함

신용카드 사기 탐지 모델

요약

- 팀구성 및 기여도 : 3명 / 25 %
- 담당 역할
    - 위험탐지 모델 개발

프로젝트 진행배경

- 최근 디지털 금융 서비스의 급속한 성장과 함께 신용카드 거래가 우리 일상의 필수적인 부분이 되었음에도 불구하고 신용사기 거래는 끊임 없이 증가함에 따라 신용카드 사기 탐지의 중요성 강조

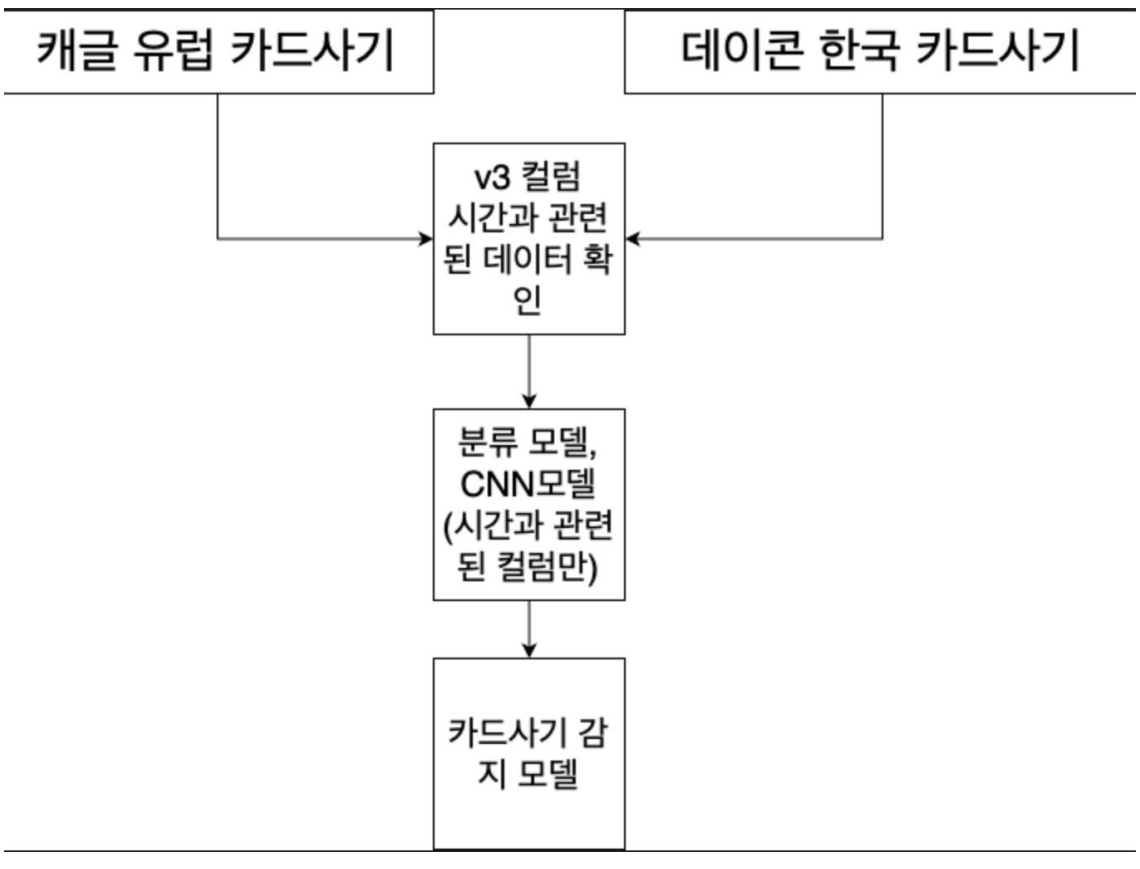
문제 개요

프로젝트명: **신용카드 사기 탐지**

수행 기간: 4주 (2024.04 ~ 2024.05)

목표: 신용카드 사기 탐지

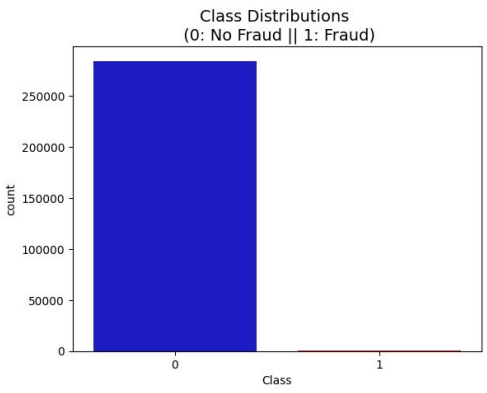
프로세스



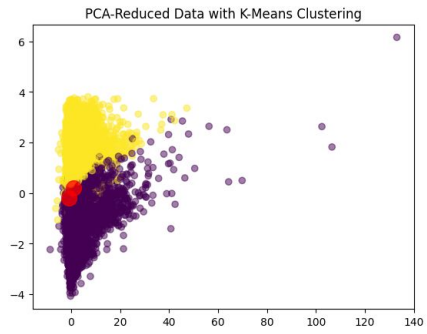
EDA

캐글 유럽 사기 데이터

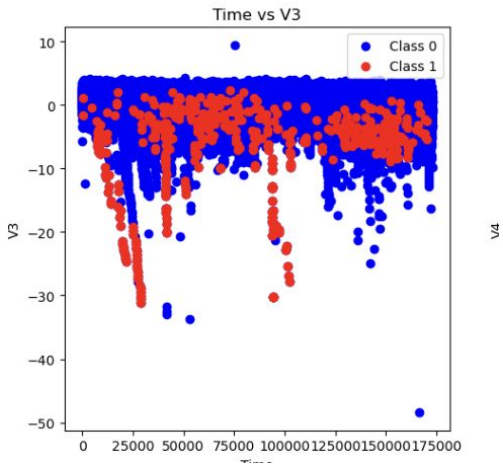
1.불균형 데이터



2.잘 분리된 클러스터링 처럼 보이지만 소수 패턴 찾기 어려움으로 인한 잘못된 클러스터링



3.데이콘 한국 카드 사기 데이터와 비교 했을때 시간과의 관련있음을 발견 v3 시간을 생각하며 모델진행

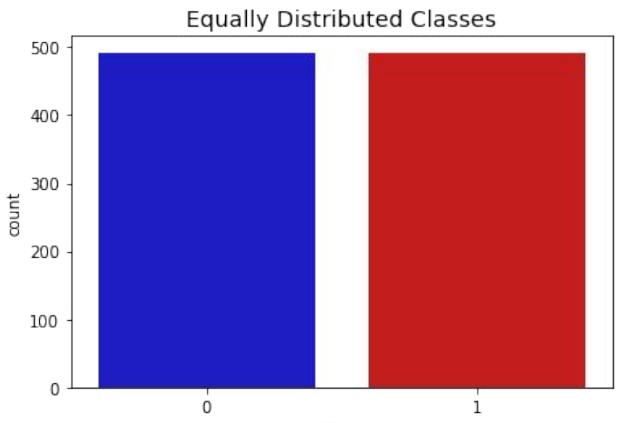


모델

1.클래스 비율 맞추기 및 정규화

StandardScaler = Amount
(평균이 0이고 분산이 1이 되도록 데이터를 표준화)

RobustScaler = Time
(이상치에 덜 민감하게 정규화)

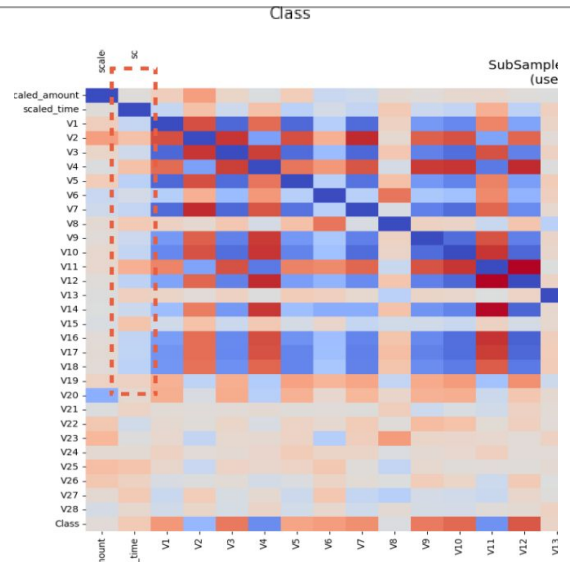


2. StratifiedKFold

LogisticRegression AUC 0.97
단순한 수학적 모델

CNN AUC 0.96
시계열 데이터와 같은 복잡한 패턴을 학습

3.시간과 관련된 컬럼만 이용
('V1','V3','V4','V5','V6','V7','V9','V10','V12','V14','V16','V17','V18','scaled_time')



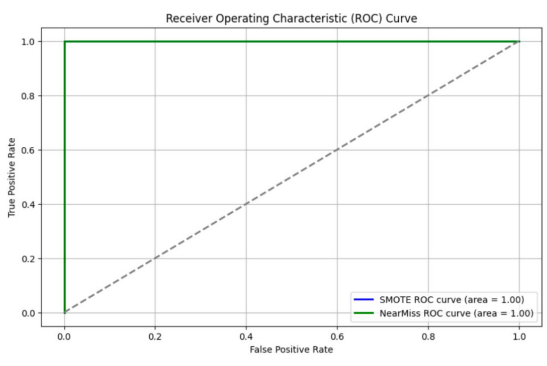
4.서포트 벡터 머신(SVM)이 AUC 0.98로 가장 높은 성능.

LogisticRegression AUC 0.97
CNN AUC 0.97
K-Nearest Neighbors AUC 0.96
Decision Tree AUC 0.91

모델 개선 시도 SMOTE, NearMiss

이는 이미 앞에서 데이터를 임의로 샘플링하여 균형잡힌 데이터셋을 만든 부분이 있어서 과적합이 일어난 것으로 생각됨

과적합 발생



Q.사기 유형이 국한 되어있다고 생각하면 전세계적으로 사기유형은 비슷하지 않을까?
> 따라서 한국 데이터를 이용한 모델로 유럽 카드 사기 감지 or 유럽 데이터를 이용한 모델로 한국 카드 사기 감지

한국 카드 사기 모델을 이용한 유럽 카드 사기 감지

Logistic Regression	CNN	knears	svc	결정 트리
auc = 0.932	auc = 0.943	auc = 0.900	auc = 0.956	auc = 0.765

유럽 카드 사기 모델을 이용한 한국 카드 사기 감지

Logistic Regression	CNN	knears	svc	결정 트리
auc = 0.986	auc = 0.988	auc = 0.967	auc = 0.991	auc = 0.950

모델 교차 검증: 전세계적으로 비슷한 유형의 사기가 진행되고 있음을 시사함 (input: v1~v28)

결론

이번 분석에서는 여러 가지 머신러닝 모델을 사용하여 신용카드 데이터셋에서 사기 거래를 감지하려고 했음. 사용된 모델은 로지스틱 회귀, k-최근접 이웃(KNN), 서포트 벡터 머신(SVM), 의사결정 나무, 그리고 합성곱 신경망(CNN)
auc 로는 svm이 높지만 향후 정확도, 정밀도, 재현율, F1 점수 등의 다양한 지표 다른 지표를 확인한 결과
로지스틱 회귀와 KNN 모델은 각각 0.94의 F1 점수와 0.94의 정확도로 최고의 성능을 보였음

두 모델을 교차해서 비교 했을때 한국 카드 사기 모델 같은 경우는 데이터 양이 적어서 다소 많이 떨어진것 같음.

느낀점

1. **데이터 전처리의 중요성**: 효과적인 스케일링 및 데이터 처리 방법은 모델의 성능에 크게 영향을 받음 `StandardScaler`와 `RobustScaler`를 사용한 정규화가 데이터셋을 훈련에 적합하게 준비하는 데 중요
2. **클래스 불균형 처리**: 원본 데이터셋은 클래스 불균형이 심함. 언더샘플링 및 계층화 샘플링과 같은 기술을 사용하여 모델이 다수 클래스에 치우치지 않도록 했음
3. **모델 비교 및 검증**: 교차 검증과 계층화 분할을 사용하여 여러 모델을 평가한 결과, 각 접근 방식의 장단점을 명확히 파악할 수 있었음
4. **CNN을 통한 딥러닝**: 이번 분류 작업에서 CNN을 구현한 결과, 전통적인 머신러닝 모델이 주로 사용되는 구조적 표 형식 데이터에서도 딥러닝 모델이 우수한 성능을 발휘할 수 있음을 확인함