

오승준 과제 및 프로젝트 포트폴리오

<https://github.com/dsjoh>

광운대학교 정보융합학부
데이터사이언스 전공

https://drive.google.com/drive/folders/11ao-RbMlie4UrO_6JuBDAoxAJ_FhGSwL?usp=drive_link

이수 과목 (1)

- **ML/DL과 관련된 과목**

- 빅데이터언어 : Python, Numpy, Pandas를 활용한 기초 데이터 분석 과목
- 데이터마이닝 : 통계 및 선형대수에 기반한 데이터마이닝을 다루는 과목
- 기계학습 : 데이터 전처리와 모델 학습에 중점을 두며, ML/DL 모델들을 다루는 과목
- 인공지능응용 : ML/DL 솔루션을 현실 세계에 적용하는 과정에서 발생하는 문제들을 처리하는 고급 방법론을 다루는 과목(데이터 불균형, 이상탐지, XAI)
- 인터랙티브 AI : MLOps, 생성 모델(GAN,VAE,Diffusion), 시계열 모델 등을 다루는 과목
- 컴퓨터비전 : OpenCV에 기반한 다양한 이미지 처리를 다루는 과목
- 비주얼컴퓨팅 : 이미지 처리 및 Image Data 기반 Deep Learning을 다루는 과목
- 텍스트마이닝 : NLP 데이터 분석 및 자연어 처리, 자연어 처리 모델들을 모두 다루는 과목

이수 과목 (2)

- **통계/수학/데이터와 관련된 과목**

- AI수학 : AI에서 사용되는 선형대수, 통계, 최적화를 다루는 과목
- 선형대수학 : 선형대수학 과목
- 실험설계및분석 : HCI와 관련한 데이터에 기반하여, ANOVA(분산) 분석을 집중적으로 다루는 과목
- 인터랙티브심리학 : HCI와 관련된 핵심 이론과 기본적인 R 사용방법을 익히고 프로젝트를 통해 실험해보는 과목
- 데이터베이스 : 전반적인 DB에 대한 세부 개념과 SQL 문법 등을 다루는 과목
- 데이터애널리틱스 : Process Mining에 대해 집중적으로 배우고, DL과 관련된 개념들을 다루는 과목

이수 과목 (3)

- **프로그래밍과 관련된 과목**

- 프로그래밍기초 : C 언어를 학습하는 과목
- 고급C프로그래밍및설계 : C++ 언어를 학습하는 과목
- 객체지향프로그래밍 : JAVA 언어를 학습하는 과목
- 자료구조 : 자료구조와 알고리즘을 학습하는 과목
- 모바일프로그래밍 : Android Studio를 기반으로 하여 안드로이드 App을 제작하는 과목
- 오픈소스소프트웨어 : 기본적인 오픈소스와 관련된 개념들과 이론들을 다루는 과목
- 오픈소스소프트웨어실습 : Git / Github의 사용법과 이를 실습하는 과목

논문 리뷰 발표 자료

- 논문 리뷰

Applicable Artificial Intelligence For Brain Disease
: A Survey

Chenxi Huang, Jian Wang, Shui-Hua Wang, Yu-Dong Zhang

인공지능응용 논문리뷰

오승준

광운대학교 정보융합학부 2019204014

Why do tree-based models still outperform deep learning on tabular data?

Leo Grinsztajn, Edouard Oyallon, Gael Varoquaux

2023. 08. 14

오승준

광운대학교 정보융합학부

S2W KRWACREATOR

광운대학교 인공지능서비스 연구실 <http://sites.google.com/view/ai-svc>

Are Transformers Effective for Time Series Forecasting?

Ailing Zeng, Muxi Chen, Lei Zhang, Qiang Xu

인터랙티브 AI 논문리뷰

Team TS1

신우현, 장한재, 노준석, 오승준

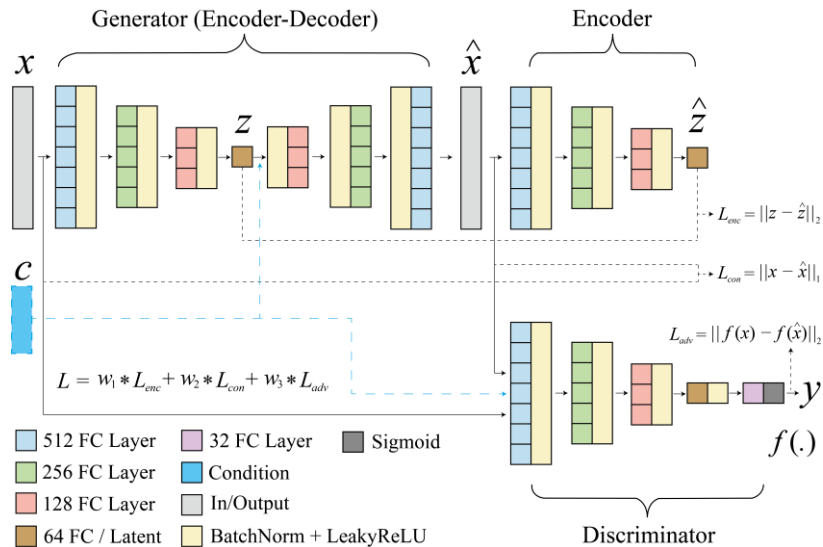
발표 자료 및 영상 링크

https://drive.google.com/drive/folders/1qVrmpkvKXC3eKWhBm3yExrnRG9wfSCHm?usp=drive_link

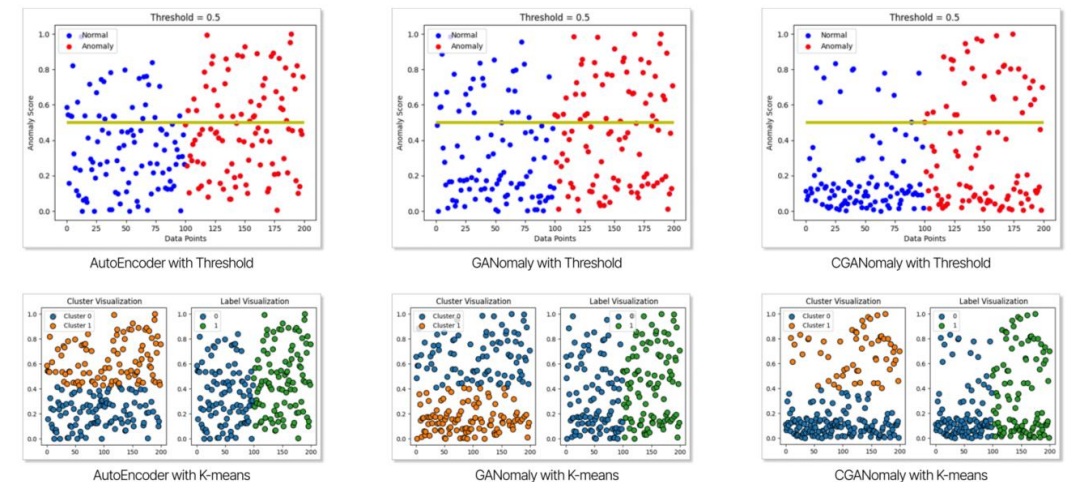
주요 프로젝트 산출물 1

인공지능응용 : 고차원 제조 공정 데이터에서의 이상탐지 (CGANomaly)

- 주어진 데이터는 1. 제품 코드, 2. 생산 라인이라는 두 가지 Categorical Feature의 조합에 따라 나머지 Feature 간의 Sparsity Pattern이 달라지는 독특한 특성을 가지고 있었으며, 특정 조합에 따라서는 절대적인 데이터 수가 충분하지 않은 경우도 있었습니다. 따라서 해당 프로젝트에서는 데이터 수가 현저히 적은 일부 조합과 데이터 수가 충분히 있는 조합을 각각 이상치와 정상 데이터로 재정의하고 다양한 이상 탐지 알고리즘을 적용해 보았습니다. 추가로, 현재 데이터셋은 중요한 Categorical Feature가 존재한다는 점을 참고하여, GAN 기반 이상 탐지 모델인 GANomaly와 Conditional GAN(CGAN)에서 영감을 얻어 현 데이터 상황을 보다 잘 반영할 수 있도록 특정 Feature를 Condition으로 줄 수 있는 CGANomaly라는 모델을 제안하게 되었고. 이렇게 제안된 모델에 대해 기존의 이상 탐지 알고리즘들과의 Comparative Study 및 Ablation Study를 진행하였습니다.



Results



발표 자료 링크

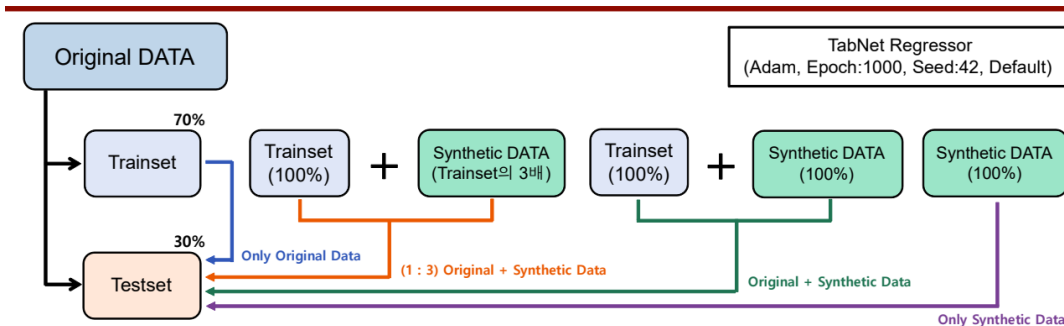
https://drive.google.com/drive/folders/1y_AAw3zdLSCAukl16vX_SAUodl5H0tA_?usp=drive_link

주요 프로젝트 산출물 2

▪ [산학과제] 서한이노빌리티 (생성모델을 활용한 시뮬레이션 데이터의 증강 및 증강된 데이터의 유효성 검증 연구)

- CGAN, CVAE 등의 생성 모델을 활용하여 설계도 제작에 사용되는 시뮬레이션 데이터를 증강하고, 증강된 데이터의 유효성을 검증하는 산학 과제에 참여하였습니다. 유효성 검증 방식에 대해서는 교수님의 조언을 참고하였으며, 그 외에 데이터 전처리부터 실험 기획, 실험, 결과물 시각화 작업 및 발표를 수행하였습니다. 데이터의 절대적인 양이 적었기 때문에 모델이 Overfit되지 않도록 신중하게 실험을 설계하였으며, 기업에서 요구했던 하나의 Input 대비 Multi-Output에 대한 결과물 및 성능 평가에 있어서도 많은 고민을 해볼 수 있었습니다.

Result

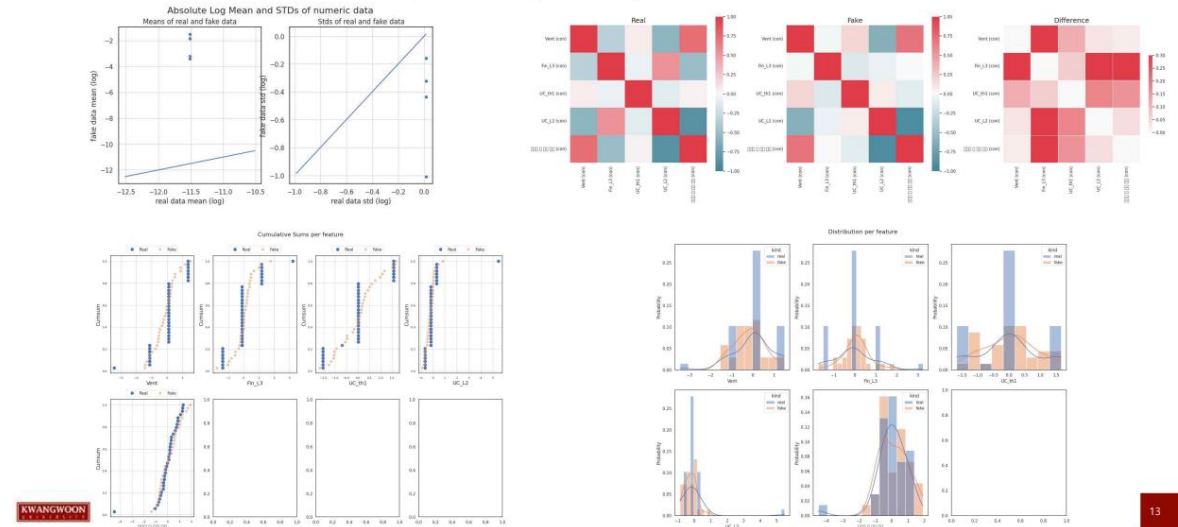


본 실험에서는 Original Data에 대해 7:3 비율로 Trainset과 Testset을 분할한 뒤, 각 Case에 대해 다음과 같은 성능을 측정하였다.

- Trainset으로 학습한 TabNet의 Testset에서의 성능 (34개)
- Trainset + Trainset의 3배 크기만큼의 Synthetic Data를 Sampling한 데이터로 학습한 TabNet의 Testset에서의 성능 (34 : 102, 34+102개)
- Synthetic Data로만 학습한 TabNet의 Testset에서의 성능 (10,000개)
- Original Data와 Synthetic Data를 모두 합친 데이터로 학습한 TabNet에서의 성능 (34 + 10,000개)

* Original + Synthetic (1:3) 비율은
이전 실험 결과를 참고하여 적당한 값을 설정

▪ Distribution & Cumulative Sums per feature (Maxstress)



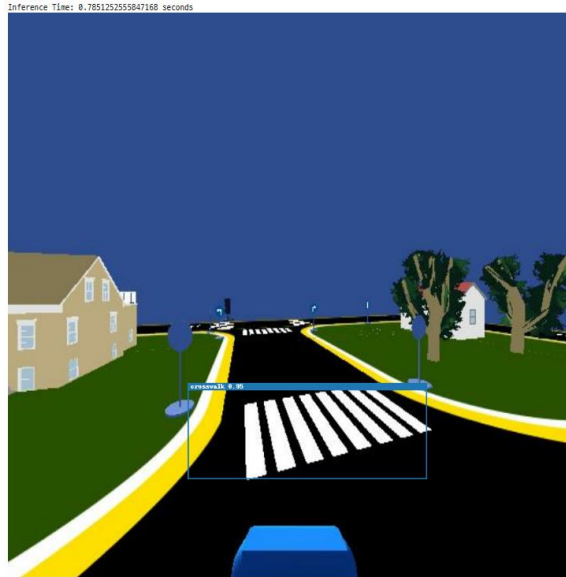
발표 자료 링크

https://drive.google.com/drive/folders/1S27RbrLwxmigoDKUMct7_eW-pyOBo16?usp=drive_link

주요 프로젝트 산출물 3

▪ [산학과제] 큐빅테크 (Unity 환경에서의 Object Detection Task)

- Unity로 구현된 도로 시뮬레이션 환경에서 객체를 탐지하고, 탐지된 정보를 기반으로 시뮬레이션 속 자율주행 자동차가 올바른 의사 결정을 내리는 것을 목표로 하는 산학 과제에서 Object Detection 파트를 담당하였습니다. 모델 크기와 추론 속도에 대한 제약이 있었으며, 성능에 대한 최소 목표치가 설정되어 있었습니다. 그러나 목표치에 도달하지 못해 고민하던 중, 모델의 추론 결과를 통해 데이터의 일관성 문제와 불균형 문제가 focal loss에 영향을 주었다고 판단하였습니다. 이를 바탕으로 모든 데이터를 Re-Labeling하고 focal loss의 Hyperparameter를 튜닝한 결과, 모델 성능을 목표치까지 향상시킬 수 있었습니다.



```
Running per image evaluation...
Evaluate annotation type *bbox*
DONE (t=0.03s).
Accumulating evaluation results...
DONE (t=0.03s).
Average Precision (AP) @[ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.586
Average Precision (AP) @[ IoU=0.50 | area= all | maxDets=100 ] = 0.871
Average Precision (AP) @[ IoU=0.75 | area= all | maxDets=100 ] = 0.697
Average Precision (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.267
Average Precision (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.644
Average Precision (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.741
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets= 1 ] = 0.568
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets= 10 ] = 0.607
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.607
Average Recall (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.272
Average Recall (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.670
Average Recall (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.750
```

실험 노트북

https://drive.google.com/drive/folders/1JC60Ve02Mg3_tsHb8VPbQ2j0d5ytQfoR?usp=drive_link

주요 프로젝트 산출물 4

■ 텍스트마이닝 : low level 챗봇 기반 서울시 관광 코스 추천 시스템(오늘 뭐하지?)

- 네이버 지도에서 제공하는 데이터와 추상적인 표현이 섞인 사용자의 답변을 임베딩하고, 데이터와 사용자의 답변 사이의 유사도를 측정하여 적절한 다음 장소를 추천하는 방식으로 관광 코스를 완성하는 로우 레벨 챗봇 개발 프로젝트를 진행하였습니다.
데이터 크롤링을 제외한 프로젝트 기획부터 메인 알고리즘, 발표까지 모두 수행하였으며, FastText에서 제공하는 임베딩된 단어들 간의 거리를 기반으로 관련성이 높은 단어를 계산하는 기능에 흥미를 느껴 이를 활용한 프로젝트를 기획하게 되었습니다.

+

데이터 전처리

<수집된 데이터 : 약 50,000개>

식당

카페

장소

술집

- 네이버 평점 3.0 이하의 장소들은 Filtering
- 대표메뉴에 (300ml, 중)등이 들어간 경우 정규화
- Geopy를 위한 도로명 주소 정규화
- 업종, 분류 항목 재분류 및 재분류
- 결측치(리뷰, 평점, 주소 등)가 너무 많은 경우 제거
- 동일한 장소 Filtering
(동일한 식당이 1F, 4F를 모두 운영)

1

import pandas as pd

2

from geopy.geocoders import Nominatin

3

from geopy.exc import GeocoderTimedOut

4

import re

5

도로명 주소를 기반으로 x, y 좌표를 생성하는 함수

6

def geocode_address(address):

7

geolocator = Nominatin(user_agent="my_geocoder")

8

try:

9

location = geolocator.geocode(address)

10

if location:

11

return location.latitude, location.longitude

12

else:

13

return None, None

14

except GeocoderTimedOut:

15

return geocode_address(address)

16

data = place

17

20

도로명 주소를 기반으로 x, y 좌표 생성

21

x_coords = []

22

y_coords = []

23

for address in data['주소']:

24

x, y = geocode_address(address)

25

x_coords.append(x)

26

y_coords.append(y)

27

28

x, y 좌표를 데이터프레임에 추가

29

data['x'] = x_coords

30

data['y'] = y_coords

1

데이터 로드 및 전처리

2

data = pd.read_csv('data/place.csv')

3

데이터 로드 후, 평점 3.0 이하의 장소는 필터링

4

data = data[data['평점'] > 3.0]

5

데이터 로드 후, 대표메뉴에 (300ml, 중)이 들어간 경우 정규화

6

data['대표메뉴'] = data['대표메뉴'].str.replace(' (300ml, 중)', '')

7

데이터 로드 후, 업종, 분류 항목 재분류 및 재분류

8

data['업종'] = data['업종'].str.replace(' 음식점', ' 식당')

9

data['분류'] = data['분류'].str.replace(' 카페', ' 카페')

10

결측치(리뷰, 평점, 주소 등)가 너무 많은 경우 제거

11

data = data.dropna(subset=['리뷰', '평점', '주소'])

12

동일한 장소 Filtering

13

data = data.drop_duplicates(subset=['주소'], keep='first')

14

도로명 주소를 기반으로 x, y 좌표 생성

15

data['x'], data['y'] = data.apply(lambda row: geocode_address(row['주소']), axis=1, result_type='expand')

16

데이터프레임에 x, y 좌표 추가

17

data['x'] = data['x'].astype(float)

18

data['y'] = data['y'].astype(float)

19

데이터프레임에 x, y 좌표 추가

20

data['x'] = data['x'].astype(float)

21

data['y'] = data['y'].astype(float)

22

데이터프레임에 x, y 좌표 추가

23

data['x'] = data['x'].astype(float)

24

data['y'] = data['y'].astype(float)

25

데이터프레임에 x, y 좌표 추가

26

data['x'] = data['x'].astype(float)

27

data['y'] = data['y'].astype(float)

28

데이터프레임에 x, y 좌표 추가

29

data['x'] = data['x'].astype(float)

30

data['y'] = data['y'].astype(float)

<도로명 주소 -> 위도, 경도>

1

도로명 주소를 기반으로 x, y 좌표 생성

2

def geocode_address(address):

3

geolocator = Nominatin(user_agent="my_geocoder")

4

try:

5

location = geolocator.geocode(address)

6

if location:

7

return location.latitude, location.longitude

8

else:

9

return None, None

10

except GeocoderTimedOut:

11

return geocode_address(address)

12

data = place

13

20

도로명 주소를 기반으로 x, y 좌표 생성

21

x_coords = []

22

y_coords = []

23

for address in data['주소']:

24

x, y = geocode_address(address)

25

x_coords.append(x)

26

y_coords.append(y)

27

28

x, y 좌표를 데이터프레임에 추가

29

data['x'] = x_coords

30

data['y'] = y_coords

1

데이터 로드 및 전처리

2

data = pd.read_csv('data/place.csv')

3

데이터 로드 후, 평점 3.0 이하의 장소는 필터링

4

data = data[data['평점'] > 3.0]

5

데이터 로드 후, 대표메뉴에 (300ml, 중)이 들어간 경우 정규화

6

data['대표메뉴'] = data['대표메뉴'].str.replace(' (300ml, 중)', '')

7

데이터 로드 후, 업종, 분류 항목 재분류 및 재분류

8

data['업종'] = data['업종'].str.replace(' 음식점', ' 식당')

9

data['분류'] = data['분류'].str.replace(' 카페', ' 카페')

10

결측치(리뷰, 평점, 주소 등)가 너무 많은 경우 제거

11

data = data.dropna(subset=['리뷰', '평점', '주소'])

12

동일한 장소 Filtering

13

data = data.drop_duplicates(subset=['주소'], keep='first')

14

도로명 주소를 기반으로 x, y 좌표 생성

15

data['x'], data['y'] = data.apply(lambda row: geocode_address(row['주소']), axis=1, result_type='expand')

16

데이터프레임에 x, y 좌표 추가

17

data['x'] = data['x'].astype(float)

18

data['y'] = data['y'].astype(float)

19

데이터프레임에 x, y 좌표 추가

20

data['x'] = data['x'].astype(float)

21

data['y'] = data['y'].astype(float)

22

데이터프레임에 x, y 좌표 추가

23

data['x'] = data['x'].astype(float)

24

data['y'] = data['y'].astype(float)

25

데이터프레임에 x, y 좌표 추가

26

data['x'] = data['x'].astype(float)

27

data['y'] = data['y'].astype(float)

28

데이터프레임에 x, y 좌표 추가

29

data['x'] = data['x'].astype(float)

30

data['y'] = data['y'].astype(float)

+

챗봇

안녕하세요! 멋진 하루를 보낼 수 있도록 도와드리겠습니다!

오늘 어디를 방문하실 계획이 있으신가요? 방문 계획을 있다면 인근 지역별 역을 알려드리겠습니다?
(가장 가까운 역을 알려주세요)

받은 분류기에서 스트레스를 받고 싶어

말씀해주신 결과를 바탕으로 다음과 같은 장소를 찾아보았습니다! 마음에 드는 번호를 알려주세요

1번

장소명: 롯데월드 아이스링크, 대표메뉴: 아이스링크

2번

장소명: 송파어린이영어도서관, 대표메뉴: 어린이도서관

3번

장소명: 롯데월드 시네마, 대표메뉴: 이탈리아음식

4번

장소명: 시각각 놀이방, 대표메뉴: 놀이방

5번

장소명: 서울시립수목원, 대표메뉴: 서울시립수목원

6번

장소명: 롯데월드 가든스테이지, 대표메뉴: 가든스테이지

7번

장소명: 강남소호호텔, 대표메뉴: 호텔

8번

장소명: 카이로 갤러리, 대표메뉴: 갤러리

9번

장소명: 롯데월드 아쿠아리움, 대표메뉴: 아쿠아리움

10번

장소명: 롯데월드 어드벤처, 대표메뉴: 테마파크

마지막으로 아직 아실다면 마무리로 술집은 어떤가요? 가고 싶은 술집에 대해 자유롭게 적어주세요
(술집 추천을 원하지 않으시면 X를 입력해주세요)

조금 출출할것같은데, 적당한 안주가 있으면 좋겠다. 라면집 추천

말씀해주신 결과를 바탕으로 다음과 같은 장소를 찾아보았습니다! 마음에 드는 번호를 알려주세요

1번

장소명: 바타, 대표메뉴: 바타

2번

장소명: 구기하루(86 LOUNGE), 대표메뉴: 바타

3번

장소명: 세컨드 스테이지, 대표메뉴: 바타

4번

장소명: 액션하우스, 대표메뉴: 바타

5번

장소명: 코코콜라, 대표메뉴: 바타

6번

장소명: 맥주클럽, 대표메뉴: 바타

7번

장소명: 만주, 대표메뉴: 바타

8번

장소명: 소위당, 대표메뉴: 바타

9번

장소명: K.PUB (케이비스트), 대표메뉴: 바타

10번

장소명: 게이밍, 대표메뉴: 바타

발표 자료 링크

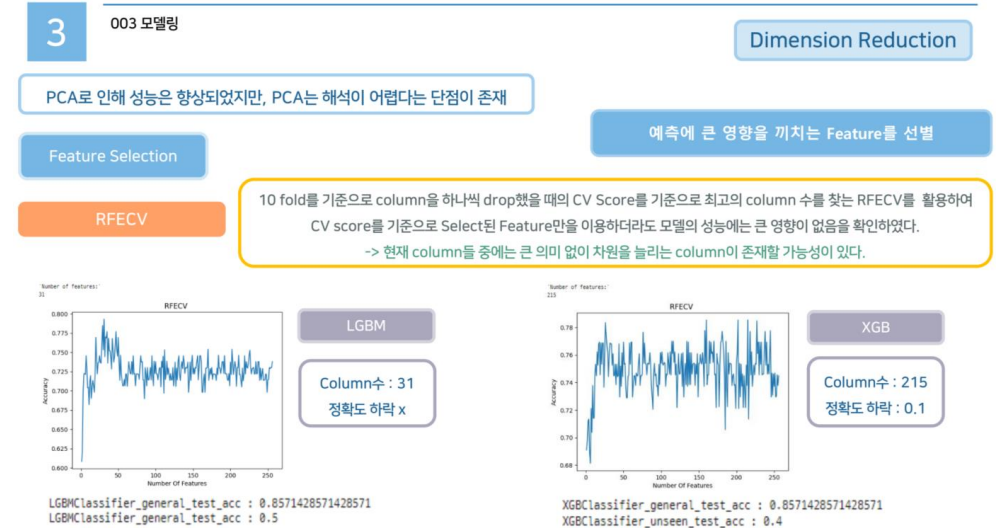
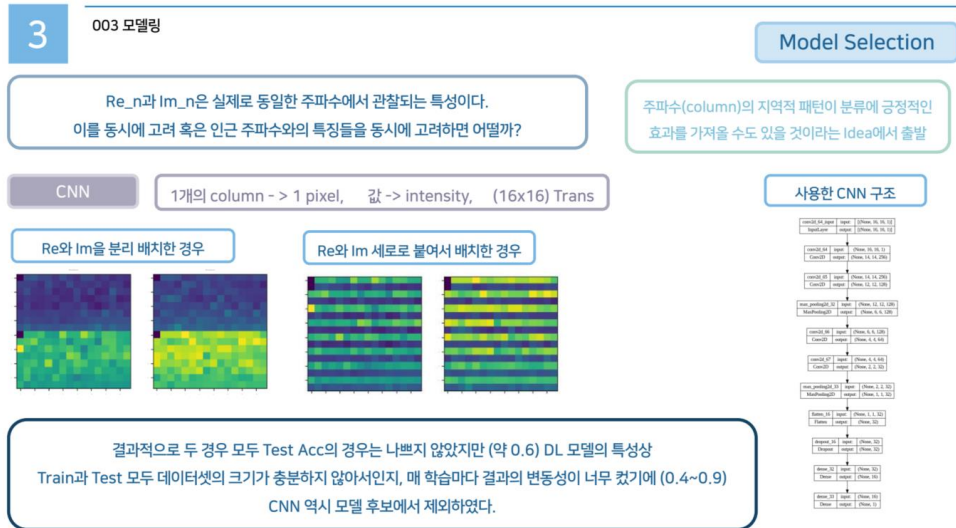
https://drive.google.com/drive/folders/1Dyb3lncsa-UialGFjA8BvtR3selOgCAP?usp=drive_link

9

주요 프로젝트 산출물 5

기계학습 : 스마트폰 센서 데이터 기반 신경 퇴행성 질환 예측

- 정상인과 파킨슨병 환자의 스마트폰의 다양한 센서들로부터 수집된 데이터에 적절한 전처리를 적용하고, 준수한 예측 성능을 보여주는 학습 모델을 구축해 본 뒤, 최종적으로 중요한 Feature를 선별하는 ML 프로젝트를 진행하였습니다. 프로젝트 기획부터 전처리, 모델링, 성능 향상을 위한 실험, 발표 등의 전체 과정을 수행하였으며, No Free Lunch 이론에 따라 최적의 성능을 위해 선택된 9개의 ML 모델 중 5개의 ML 모델을 선별하고, 각 모델마다 이상치 처리를 한 경우와 하지 않은 경우 모두 각각 3개의 Scaler를 적용하여 비교 실험을 진행하였습니다. 이후에는 각 경우에 대해 Feature Extraction/Selection 등을 시도하며 성능 변화를 관찰하였고, 이를 통해 얻은 결론과 Tree based model들에서 제공하는 Feature Importance, 마지막으로 SHAP value를 활용하여 공통적으로 중요도가 높은 Feature들을 선별하고, 군집화 분석을 통해 데이터에 대한 새로운 인사이트를 도출 및 이에 기반한 활용 방안을 구상해보는 프로젝트를 수행하였습니다.



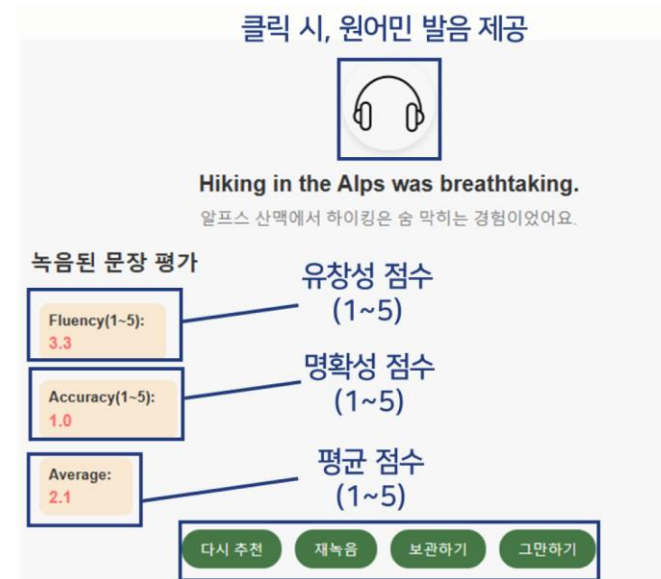
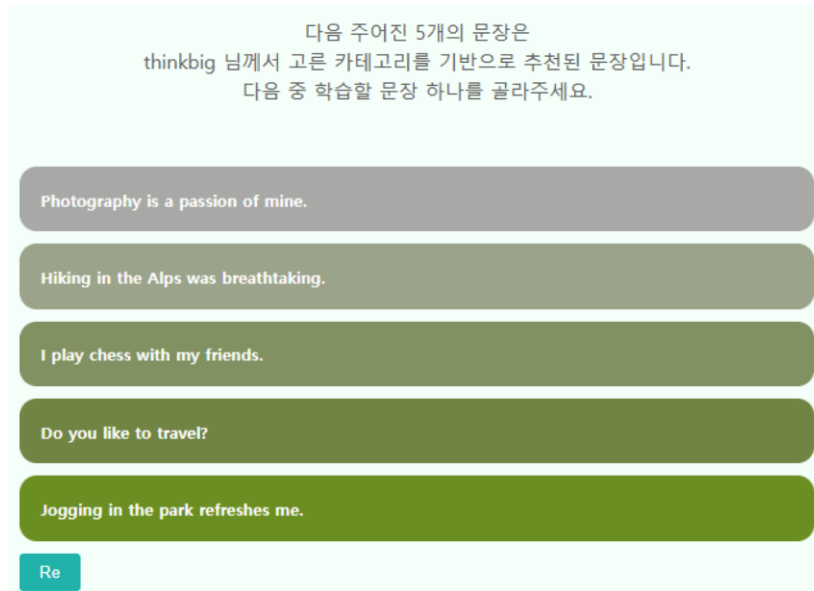
발표 자료 및 영상 링크

https://drive.google.com/drive/folders/1EBtxOAglkew-vELZijlSTlfeUOLIFesg?usp=drive_link

주요 프로젝트 산출물 6

■ 졸업작품 : 사용자 발음 취약 음소 분석과 대규모 언어 모델을 활용한 멀티모달 기반 문장 추천 시스템

- 사용자의 문장 발음에 대해 딥러닝 기반의 Score를 계산하여 유창성 점수를, 음성 인식 결과를 CMU 음소 문자열로 변환한 결과에 대해 Score를 계산하여 명확성 점수를 계산하고, 취약 음소를 추출한 뒤 이 모든 정보를 LLM의 Prompt로 전달하여 사용자의 취약 발음 반복 훈련을 유도하여 최종적으로 사용자의 발음 학습을 돕는 문장 추천 시스템을 개발하였습니다. 이 과정에서 각 모델을 API화하여 배포하였으며, 팀의 리더로서 전체 서비스 기획은 물론 백엔드 서버, 프론트엔드 파트 개발자분들과 함께 소통하며 협업하는 기회를 얻을 수 있었습니다.



발표 자료 및 영상 링크

<https://youtu.be/yWXYF2xx75g>

주요 프로젝트 이외 과제물 1

■ 기계학습 : 군집화와 의사결정나무 관련 실험 보고서

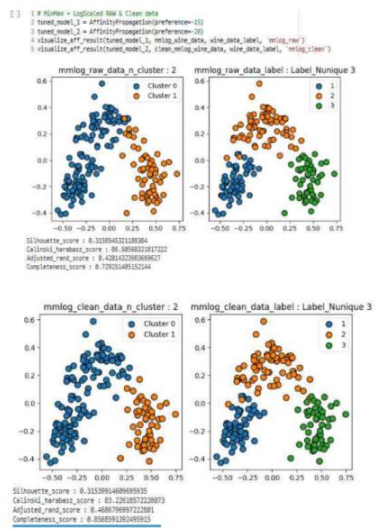
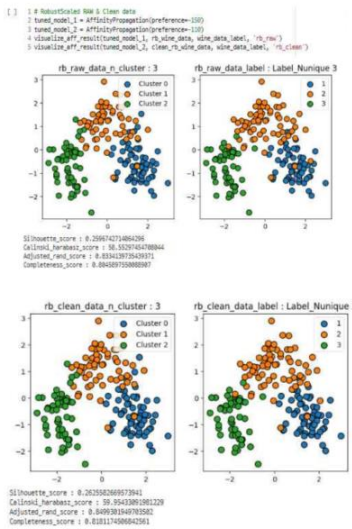
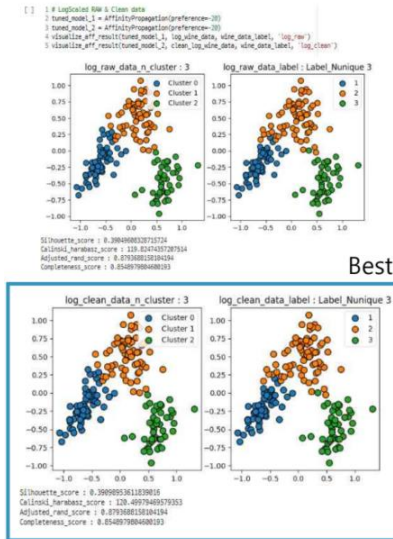
2

2.4-4 AffinityPropagation 군집화 분석 및 가시화

- Affinity LogScaled Raw & Clean data -

- Affinity RobustScaled Raw & Clean data -

- Affinity MM+LogScaled Raw & Clean data -



3

3.2 의사결정나무 구축 및 실험비교

- Pruning 후 성능 평가 -

Metric	before Prune	After Prune
Train_acc	1.0	0.956
Test_acc	0.893	0.905
Cross_val_score	0.894	0.906
Macro_acc	0.905	0.905
Macro_recall	0.910	0.914
Macro_precision	0.918	0.918
Macro_f1	0.916	0.916
Micro_f1	0.900	0.905

Prune 이후, Train의 정확도는 감소, test와 cross_val_score는 증가함으로써 train data에 대한 모델의 overfitting이 감소하고 general 한 모델의 성능이 올라갔음을 알 수 있다.

불순도를 기반으로 지도 학습을 하는 의사결정나무 모델의 특성상 overfit이 발생하기 쉬우며, 이는 prune을 통해 어느정도 통제할 수 있다. (그러나, 과한 prune은 설명력의 감소로 모델의 성능 자체를 저하시킬 수 있다. [underfitting])

보고서 링크

https://drive.google.com/drive/folders/1uibHOZt6eKPAi2k3vnKDNGi-FziwEsuB?usp=drive_link

주요 프로젝트 이외 과제물 2

- 데이터마이닝 : 선형회귀와 로지스틱 회귀 & Shrinkage Models 관련 실험 보고서

4

4.2 예측 정확도 향상

ElasticNet Hyperparameter Tuning

[illegible]

Grid Search 방식으로 Hyperparameter를 Tuning
기준 : Validation
[0.07, 0.05, 0.03, 0.01, 0.005]
[0.1, 0.3, 0.5, 0.7, 0.9]

```
alpha = 0.01
l1_ratio = 0.5
```

Valid accuracy: 0.9178082191780822
Valid recall: 0.926
Valid precision: 0.862
Valid f1-score: 0.893

모든 Shrink Classifier가 sklearn에 구현되어 있는 것은 아니기에, 일부 Classifier의 경우는 다른 Classifier의 Option을 활용해 간접적으로 구현하였다.

또한 Hyperparameter Tuning을 진행하는 과정에서 ElasticNet은 hyperparameter 조합의 차이만으로 특정 Metric이 부분적으로만 다른 Hyperparameter 조합의 모델들보다 크게 개선되는 경우가 존재한다.

이를 통해 Hyperparameter Tuning이 단점으로 여겨지는 ElasticNet의 특징을 다시 한 번 확인할 수 있다.

5

5.2 선형회귀 결과 해석

- Linear Regression Model Evaluation (MAE / MSE) -

Train의 MAE / MSE

```

1 from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
2
3 # Train dataset에 대한 학습 평가
4 x_train = train.drop('price', axis=1)
5 y_train = train['price']
6 x_train_rm = sm.add_constant(x_train) # 0 첫 번째 위치의 'const' 값을 추가함
7
8 # 예측 결과 출력 (train)
9 y_pred_train = model.predict(x_train)
10 mse_train = mean_squared_error(y_train, y_pred_train)
11 mae_train = mean_absolute_error(y_train, y_pred_train)
12
13
14 print("Training made:", mse_train)
15 print("Training MAE:", mae_train)
16
17
18 Training made: 1239.18469997807
19 Training MAE: 30.984314977110687

```

Valid의 MAE / MSE

```

> x_valid = dataset.x에 대한 샘플링
> x_valid = valid.drop('price', axis=1)
> y_valid = valid['price']
> x_valid_sm = sm.add_constant(x_valid) # 첫 번째 위치에서
5
> 예측 및 성능 평가 (Validation)
> y_pred_valid = model.predict(x_valid_sm)
> mse_valid = mean_absolute_error(y_valid, y_pred_valid)
> mae_valid = mean_squared_error(y_valid, y_pred_valid)
10
11
12 print("Validation MAE:", mae_valid)
13 print("Validation MSE:", mse_valid)

Validation MAE: 88175.54319930089
Validation MSE: 104347621114.30695

```

Test의 MAE / MSE

```
1 # Test dataset에 대한 성능 평가
2 X_test = test.drop('price', axis=1)
3 y_test = test['price']
4 X_test_sm = sm.add_constant(X_test)
5
6 # 회귀할 성능 평가 (Test)
7 y_pred_test = model.predict(X_test_sm)
8 mse_test = mean_absolute_error(y_test, y_pred_test)
9 mse_test = mean_squared_error(y_test, y_pred_test)
10
11 print("\nTest MAE:", mse_test)
12 print("\nTest MSE:", mse_test)
```

Test MAE: 2241.533548130708
Test MSE: 11362172.952448729

LR 모델은 전체 Full Dataset 중 Train만을 학습했기 때문에 Train의 MAE / MSE는 Trainset에 대해서 예측된 Y와 실제 Y 간의 차이를 최소화
최적화 한 결과이다. 따라서 학습한 LR 모델은 Full Data의 일부분인 Train에 대해서만 Local Optim한 Solution을 도출했다고 이야기 할 수 있다.

그러나, 마찬가지로 Full Data의 일부에 해당하는 Valid와 Test에 대한 MAE / MSE를 비교하면 Train에 비해 최소 2배에서 최대 10배 이상 차이는 모습을 보이는데, 이는 완전 해석적이고 이머지는 결과이다. 원인으로서는 Train이 Full Data의 분포를 반영하지 못했거나, Valid/Test가 이상치를 과하게 포함했거나 등 모델 자체가 Train에 과적합된 상황이라고 예상할 수 있는데,

앞서 확인했듯이 현재 모델은 다중 공산성 및 데이터 분포에 의한 Overfitting이 존재했을 가능성이 높으므로 추후 Shrinkage model을 통해 이를 일부 해결할 수 있다.

추가적으로, Data의 특성 자체는 Valid보다는 Test가 Train과 조금 더 유사한 것을 예상할 수 있다.

보고서 링크

https://drive.google.com/drive/folders/1l9onhGF9VX1TiTeEo6ue_93Z_BXlRBiN?usp=drive_link

주요 프로젝트 이외 과제물 3

- 인터랙티브 AI : WandB를 활용한 Hyperparameter Tuning 실험 보고서

Hyperparameter

- Hyperparameter Tuning

WandB의 Sweep을 활용하여 Hyperparameter Tuning을 진행한다.

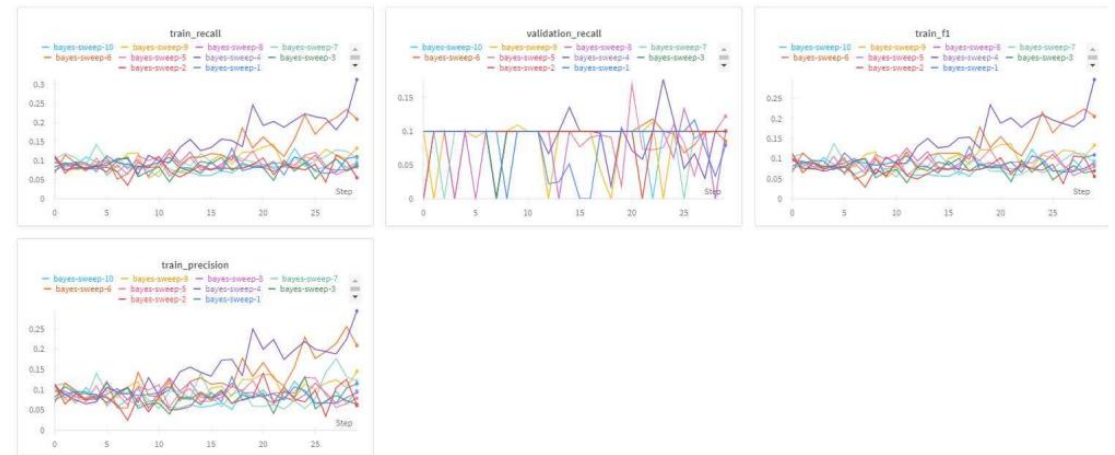
1. Random Search를 기반으로 한 10개의 Sweep을 진행 (Random Sweep)
2. Bayesian Optimization을 기반으로 한 10개의 Sweep을 진행 (Bayes Sweep)
3. 이전 20개의 Sweep 중 가장 성능이 좋았던 Sweep의 HyperParameter를 기준으로, Random Seed만을 바꾸어가며 10개의 Sweep을 진행 (Seed Sweep)

총 30번의 Sweep을 거친 Hyperparameter Tuning

4

Bayes Sweep

- Bayes Sweep



14

보고서 링크

https://drive.google.com/drive/folders/1-mE77gv4dy8ap9enEVIInm6TK9u2W1wOs?usp=drive_link

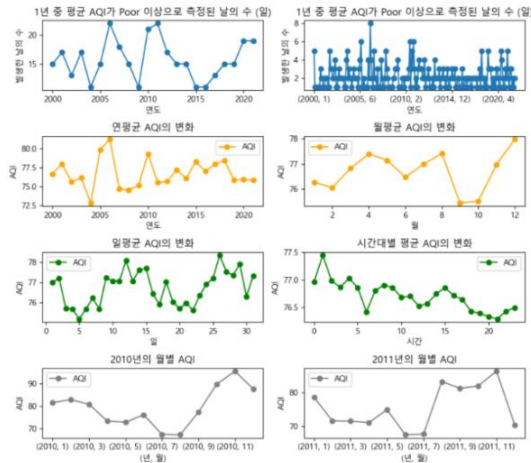
주요 프로젝트 이외 과제물 4

빅데이터언어 : 서울시 대기환경 분석 프로젝트 보고서

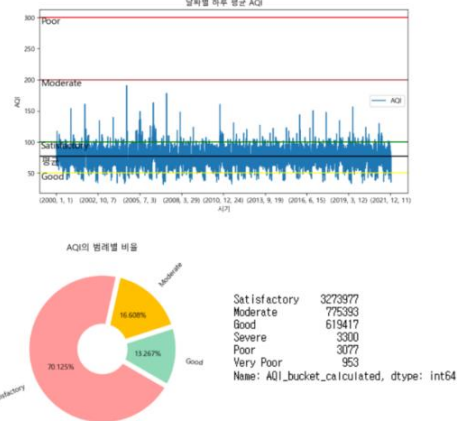
3

003 데이터 분석

시계열 패턴 및 경향 분석

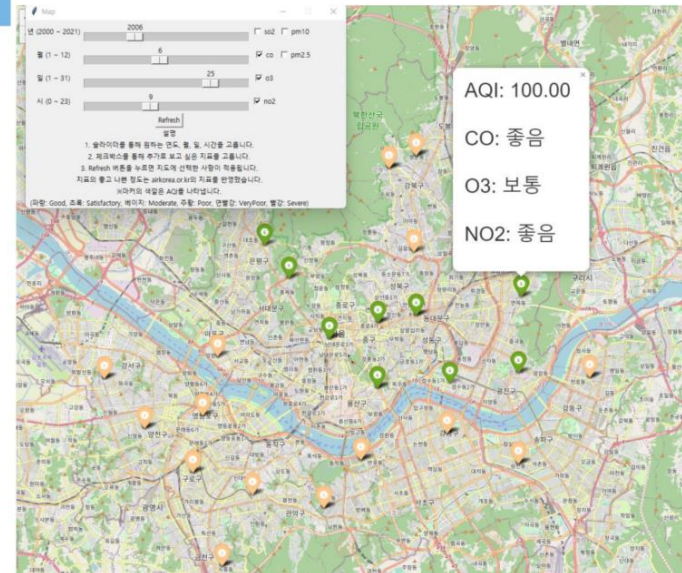


서울시의 공기는 대부분 Satisfactory



4

004 GUI 개발



Tkinter

- ▶ 슬라이더를 통해 유저에게 [년, 월, 일, 시간]을 입력 받는다.
- ▶ 체크박스를 통해 추가로 보여줄 데이터를 입력 받는다.

Folium

- ▶ 지도와 마커를 구현해서 HTML파일로 저장해준다.
- ▶ AQI 수치에 따라 마커의 색깔을 바꾸어 시각효과를 준다.
- ▶ 마커 클릭 시 Popup을 통해 추가적인 정보를 준다.

Selenium

- ▶ 저장된 HTML파일을 Chrome으로 열고 새로고침 해준다.

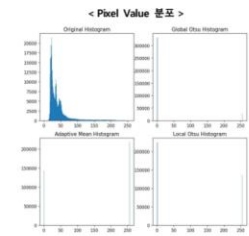
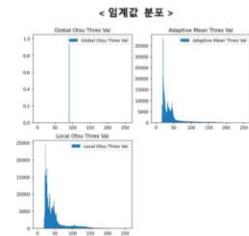
코드 및 보고서 링크

https://drive.google.com/drive/folders/1j5bqct7XCX27hgFzG0038qZRykuDCNSI?usp=drive_link

주요 프로젝트 이외 과제물 5

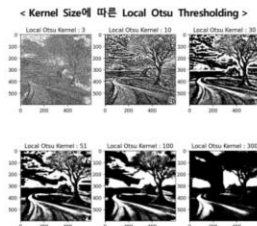
■ 비주얼컴퓨팅 : OpenCV를 활용한 과제 & 보고서

② 실험 1-2



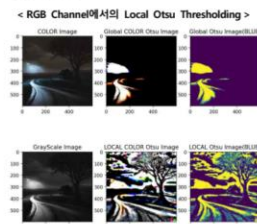
Threshold 방법에 따른 임계값과 Pixel Value 분포를 보면 Global Otsu Thresholding은 단일 임계값을, 나머지 두 Thresholding은 다양한 임계값을 갖는 것을 알 수 있다. 특히 지역적인 임계값은 Local Otsu Thresholding의 경우가 Adaptive Mean Thresholding의 경우보다 다채롭게 결정된 것을 알 수 있으며, 흑백 Pixel 분포를 참고하면 Adaptive Mean Thresholding의 경우는 나머지 Otsu Thresholding보다 대체적으로 흰색으로 Mapping된 Pixel의 수가 많다는 특징이 있다.

③ 실험 1-3



Local Otsu Thresholding에서 Kernel Size가 작을수록 Noise가 증가하고 Kernel Size 만큼의 주변 Pixel 간 값 차이가 크기 않아 Kernel Size 만큼의 동일한 값들로 채워진 Pixel block들이 자주 관찰된다. 반대로 Kernel Size가 커질수록 점점 Global Otsu Thresholding과 유사한 결과가 나타난다. 따라서 Kernel Based Thresholding에는 적절한 Kernel Size가 중요하다는 것을 확인할 수 있다.

④ 실험 1-4



GrayScale에서의 Thresholding뿐만 아니라 RGB Channel로 split 후 각각의 Color Channel 별로

③ 실험 3



Image Pyramid Blending 과정에서 Max Pyramid level을 달리했을 때의 결과를 나타내 보았다. 그 결과 Max level에 따라 두 이미지 간의 경계면과 전체적인 이미지의 백색 정도 사이의 Trade-off가 관찰되었으며, 이는 앞선 실험 결과와도 마찬가지로 Laplacian Pyramid의 level이 높아질수록 더욱 세세한 detail(Edge)들이 Detection되고 더해지는 과정에서 이러한 일부 noise 내지는 high frequency 영역들이 강조된 상태가 Gaussian Pyramid의 Upsampling 과정에서 영향을 미친 것으로 해석할 수 있었다. 따라서 적정 수준으로 판단되는 Max Level 5의 경우를 중심으로 이렇게 뚜렷하게 나타나는 Reconstructed Image의 Contrast를 올리기 위해 몇 가지 추가적인 시도를 해 보았다.

III. Advanced Topic

3.1 Color Bleeding

Image Reconstruction 과정을 살펴보면, 재구성된 이미지의 high frequency detail은 Laplacian Pyramid Image로부터 전달받으며, 이후 Gaussian Pyramid Image에서 UpSampling 과정을 거치게 된다. 그러나 이때의 UpSampling은 저해상도의 Color 정보를 담고 있는 Pixel로부터 고해상도의 넓은 영역을 이루는 Pixel들의 Color 정보를 Reconstruction 하게 되는데, 이 과정에서 인접한 Pixel 간의 값들이 혼합되어 이미지가 전반적으로 뿌옇게 나타나는 (Contrast가 낮아지는) Color Bleeding 현상이 나타나게 된다.

본 실험에서는 보다 완성도있는 Blended Image를 구해보고자 Pyramid Max Level 5에서 Blending된 이미지에 대해 Contrast를 높일 수 있는 몇 가지 추가적인 시도를 해 보았다.

3.2 CLAHE



광도 L을 따로 분할하는 LAB 색공간으로 변환 후 광도 L에 대해 파라미터를 clipLimit=0.6, tileGridSize=(2,2)로 두고 CLAHE를 적용한 결과는 다음과 같았다. 뿌옇게 된 정도가 조금은 줄었으며, 확실히 어두운 부분들이 일부 강조되면서 이미지의 Contrast가 높아진 모습을 관찰할 수 있었다.

3.3 Gamma Filter



④ 남은 Matching Point들에 대한 Homography 계산

앞선 과정에서 찾은 Point들을 기반으로 `cv2.findHomography`를 이용하여 Homography를 계산한다. 이때의 알고리즘은 RANSAC을 사용하며, `ransacReprojThreshold`는 5.0으로 설정한다. (데이터가 아웃라이아에 해당하는지 판단하는 Threshold)

```
Left + Mid Homography :
[[ 6.33567656e-01, 1.74388548e-01, 1.53985138e+02],
 [-2.62112947e-01, 8.95673995e-01, 3.87339644e+01],
 [-6.48171759e-05, -1.56609486e-05, 1.48000000e+00]]

Mid + Right Homography :
[[ 2.71853652e-01, -2.40278364e-02, 4.19049595e+02],
 [-2.49980122e-01, 8.23382262e-01, 2.42323473e+01],
 [-1.84254884e-03, -6.47579164e-05, 1.00000000e+00]]
```

< 이미지 조합별로 구해진 Homography >

⑤ Homography를 바탕으로 이미지를 사영 변환

mid 이미지를 중심으로 좌우로 left와 right 이미지를 붙이는 것이 자연스럽기에, left와 right 이미지를 다량 때는 두 이미지를 flip 상태에서 left 이미지에 대해 사영변환을 진행하고 Stitching된 뒤, 다시 flip하는 과정을 거쳐서 도면처럼 Stitching된 mid 이미지와 left 이미지에 대해 right 이미지가 Homography를 구하고 다시 사영변환을 가하면서 그 결과 세 이미지를 적절하게 stitching할 수 있게 된다.

따라서 Homography에 의해 변환된 image는 다음과 같으며, 각 변환된 이미지와 mid 이미지 간의 교차 Mask는 다음과 같다.



⑥ 변환된 이미지를 Stitching

결과적으로 앞선 모든 과정을 거쳐 변환된 image들을 stitching한 결과는 다음과 같다.



적절한 사영변환으로 인해 회전 및 밝기 변화가 있음에도 불구하고, 적절한 특징점을 중심으로 이미지가 잘 Stitching된 것을 확인할 수 있다. 그러나, right 이미지 같은 경우는 왜곡이 발생하였으며, left 이미지 같은 경우는 mid 이미지와의 높이 차이로 인해 일부 정보가 손실된 것을 확인할 수 있다.

코드 & 보고서 링크

https://drive.google.com/drive/folders/1j9yXabZSrrr9MrSlau-ddz0OIAIWmZ0Z?usp=drive_link

주요 프로젝트 이외 과제물 6

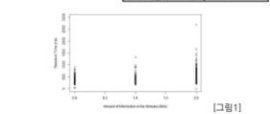
■ 인터랙티브심리학 : Choice Reaction Time & Sternberg Test(HCI) 관련 프로젝트 보고서

3.2 분석 방법

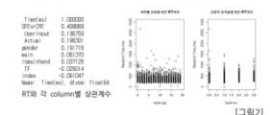
① 데이터 특성 확인

본격적인 데이터 분석에 앞서, 우선 SRT-CRT 실험으로부터 도출된 데이터들의 분포를 살펴볼 필요가 있었기에 RStudio의 plot을 활용하여 각 실험별 RT(Reaction Time) 분포를 scatter-plot으로 확인해 보았다. 이때, scatter-plot에서 X축은 독립변인에 해당하는 실험의 종류, 즉, SRT와 CRT2-2, CRT4-4를 구분짓는 선택지의 개수로 지정할 필요가 있었고, 이후 Hick's Law와의 결과 비교를 위해 선택지의 개수를 정보량으로 labeling하게 되었으며, Y축은 종속변인에 해당하는 Reaction Time(이하 RT)으로 지정하게 되었다.

	SRT-CRT	정보량 I(bits)
SRT(선택지 1개)	0.0	
CRT2(선택지 2개)	1.0	
CRT4(선택지 4개)	2.0	



[그림1]을 참고하면 각 실험별로 960개의 RT값이 찍혀있는 모습을 볼 수 있고, 전반적으로 정보량이 증가할수록 RT값이 증가하는, 대체로 양의 선형적인 관계를 나타내는 것을 확인할 수 있었다. 그러나, 각각의 실험 내에서 RT값의 편차가 크게 나타나고, CRT4-4(2.0bits)에서 혼자 유난히 RT값이 높게 나타난 지점이 존재하는 등 확인해 볼 만한 부분들이 아직 남아있었기에 추가적으로 데이터 분석을 진행하였다.



우선, 기본적으로 RT값과 추후 실험 분석을 위한 변인들을 제외한 나머지 요인 간의 상관관계를 구하였고 이 과정에서 RT값은 자국의 개수를 제외한 어떠한 변인에도 그 자체로는 큰 규모의 상관관계를 갖지는 않으며, 복합적인 요인에 의해 발생하는 편차이기에 [그림2]와 같이 RT값은 어떠한 요인에도 편차가 크게 나타나는 모습을 확인할 수 있었다. 또한, 실험적으로 500ms의 경우는 0.58에 해당하기에 지금과 같은 편차는 충분히 고려될 수 있다고 판단하였고, 이에 따라 추후

② 데이터 전처리

앞선 분석으로 [그림1]에서의 편차 문제는 이해할 수 있었지만, 여전히 총 2880개의 데이터에서 숫자는 작고 RT값이 유난히 튀는 몇 개의 데이터가 회귀모델의 Generalization을 감소시키는 outlier에 해당하는지 확인해 볼 필요는 있었다. 따라서 outlier의 후보인 RT값이 100ms이하이거나, 1500ms이상인 경우의 데이터를 [그림3]과 같이 따로 분류하였다.

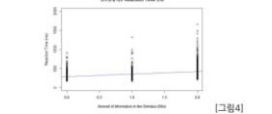
index	id	trialno	case	gender	age	sex	outlier	observed	actual	comment
1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9	9	9	9
10	10	10	10	10	10	10	10	10	10	10

outlier의 후보들은 [그림3]과 같이 3개의 CRT2-2 데이터와 3개의 SRT 데이터로 구성되어 있었다. 그중 4번 데이터는 RT값이 0으로써 정확한 실험이 진행되지 않은 명확한 outlier임을, 1~3번 데이터는 0번, 3번, 8번 case 실험자의 SRT와 평균 RT값보다 해당 데이터의 RT값이 현저히 작게 나타나므로 이 역시 outlier에, 5번과 6번 데이터의 경우는 실험자가 단순히 경음을 끄는데 시간이 오래 걸렸을 뿐이지만 6번의 경우 전체적인 CRT4-4의 RT값들과 혼자 너무 큰 차이를 나타내고 있어 회귀모델의 일반성을 위해 outlier로, 5번은 남독할 수 있는 편차 범위 내의 값이라고 판단하여 outlier 처리를 진행하지 않았다. [CRT2-2 데이터의 경우 역시 RT값이 1342ms라는 outlier 의심 데이터가 존재하지만, 남독 가능한 편차 범위 내의 RT값이기에 남겨두었다.]

3.3 데이터 분석 결과

① 실험 1-1 [SRT-CRT 데이터로부터의 Hick's Law 검증]

앞선 outlier가 처리된 데이터를 기반으로 새로운 scatter-plot과 단일선행회귀모델을 도출하여 [그림4]와 같은 결과를 얻을 수 있었다.



이는 기존의 선택지(정보량)가 증가할수록 Reaction Time도 증가한다는 Hick's Law의 결과

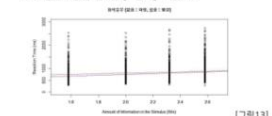
이래의 두 모델 모두 정보량의 증가에 따른 RT값의 증가를 나타내는 Hick's Law와 유사한 양상을 보이는 양의 비례관계를 만족하며, 모델과 계수의 p-value가 0.05보다 작게 나타나므로 통계적으로 유의하다고 결론지을 수 있다.

또한, 전체적인 개량에서는 잘 드러나지 않지만 [그림12-1]과 같이 두 모델을 부분 확대하면 전반적으로 피실험자가 No에 대한 응답을 할 경우는 'Yes'에 대한 응답을 할 경우보다 같은 정보량에 대해 RT값이 전반적으로 크게 나타나고 있으므로 해당 실험의 결과는 기존의 Sternberg's Classical Test의 결과와 일치하는 것을 확인할 수 있다.



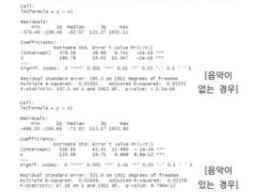
② 실험 2-3 [음악 유무에 따른 회귀모델]

이번 실험에서는 피실험자가 조용한 상황에서 Sternberg 실험을 진행했을 때(무음악)와 음악이 SSEARIM AntiFragile, 이때 반응 : 40을 들으며 실험을 진행했을 때(음악)에 도출되는 회귀모델을 통해 주변 환경이 신중한 경우에는 주변 환경이 조용한 경우보다 집중력이 떨어져 같은 정보량에 대해 RT값이 대체적으로 크게 나타날 것이라는 가설을 입증하고자 하였고, 이에 따른 회귀모델을 도출한 결과는 [그림13]과 같았다.



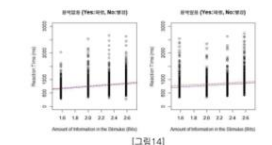
예상했던 대로, 음악을 들으며 실험하는 경우는 그렇지 않은 경우보다 같은 정보량에 대해 RT값이 대체적으로 크게 나타나는 모습을 보여줬다. 또한, 각각의 모델과 계

한 가지 독특한 점은 정보량이 증가함에 따라 음악을 들으며 실험하는 경우와 음악을 듣지 않으며 실험하는 경우에 대한 차이가 [그림13-1]과 같이 감소한다는 점이었다. 이러한 결과는 나타나는 이유에 대해서는 크게 피실험자가 실험이 진행됨에 따라 음악을 듣는 환경에 적응했거나, 제공되는 숫자 세트의 수가 줄어들수록 피실험자가 음악보다는 숫자 양에 집중하는 경향이 나타난다거나 등의 이유가 원인이 되었음을 예상할 수 있다.



③ 실험 2-4 [음악 유무와 Sternberg's Classical Test]

마지막으로는 음악 유무와 피실험자의 대답이라는 두 변인이 모두 고려된 단순선행회귀모델을 도출해 보았으며, 그 결과는 [그림14]와 같았다.



도출된 결과는 모든 모델이 계수와 모델의 p-value가 0.05보다 작게 나타나면서 Hick's Law의 결과와 유사한 양상을 나타내는 동시에 앞선 두 실험의 결론을 모두 담고 있었다. 기본적으로 피실험자가 No라고 대답하는 경우는 Yes라고 대답하는 경우보다 같은 정보량에 대해 RT값이 조금이라도 더 크게 나타나서 음악이 있는 경우는 음악이 없는 경우보다 큰 RT값을 나타내는 기존의 Sternberg's Classical Test의 결과와도 일치하면서 다시 한번 Sternberg's Classical Test의 유효성을 검증하였다.

구분	RT=a+bI (a, b)	a와 b의 신뢰도(t-value)	p-value
음악유무=Yes	(386, 177)	***	8.23e-12
음악유무=No	(373, 194)	***	4.618e-14
음악유무=Yes	(490, 143)	***	9.135e-08

보고서 링크

https://drive.google.com/drive/folders/11uBvRi6gCpCbvOqCCHXpRkaqNuphEyxl?usp=drive_link