정국호 | 데이터 분석 & AI 포트폴리오

프로젝트

1. 기후에 따른 전력량 분석 및 예측 모델

   활용 기술: Python

2. 해양 어획량 분석 프로젝트

   활용 기술 : Python

3. 표정 연습 애플리케이션

   활용기술 : Python , Flask , Flutter

4. 감정 기반 컨텐츠 추천

   활용기술 : Python, Streamlit

5. AI 기반 뉴스 요약 및 토픽 분석 서비스

   활용 기술 : python,Flask,Html,transformers

6. 보이스피싱 AI 탐지 서비스

   활용 기술: Langchain,Langgraph,RAG,faiss,openaiapi
   cohere

해양 어획량 분석 프로젝트

- **프로젝트 주제**

  해양 기후 요소가 어획량에 미치는 영향을 파악하고
  예측 모델을 구축

- **프로젝트의 어려움**

  1. 어획량 데이터의 심한 변동성으로 분석 신뢰도 확보 어려움.
  '어획률' 지표 변경 시도했으나 변동성 완전 해소는 어려움.

  2. 데이터 복잡성으로 다양한 예측 모델 적용에도 RMSE/MAE
  지표 높아 예측 정확도 현저히 낮음.

- 프로젝트 일정 : 2025.04.30 ~ 2025.05.16
- 프로젝트 인원 : 4명
- 개인 기여도 : 70% (코드 작성, 로드맵작성, 가설검증)
- 관련 스킬 : 딥러닝, 머신러닝, 시계열
- 세부스킬 : seaborn, matplotlib, CNN, LSTM, XGB

- **다양한 데이터의 접근**

  1. 실제 위성 사진을 통한 예측

  2. 해양사이트에서 제공되어지는 위성 사진 내 수치데이터

  3. 기상청에서 제공되는 수치데이터

  링크 : https://drive.google.com/drive/folders/1sbcTbpWap7L0V6zI9bR7Hhy7_Q3AeFeW

- **다양한 방법의 특성 전처리**

  1. CNN을 통해 시계열 패턴 파악

  2. 도메인 기반 특성 선택

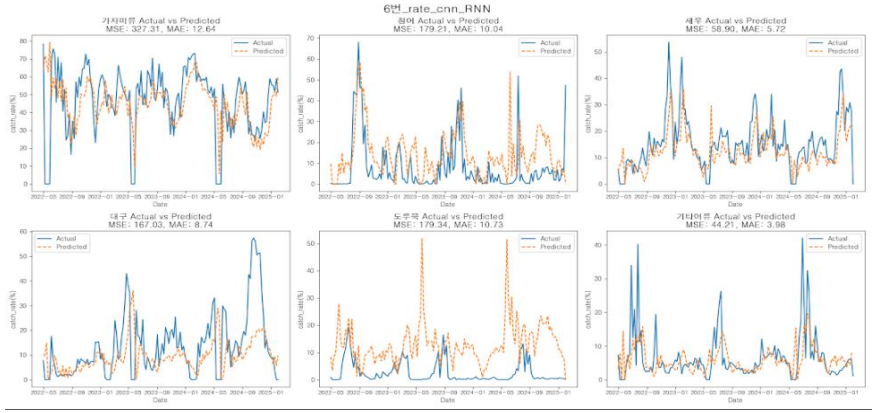
  3. PCA 기반 차원 축소

  4. 단계적 선택법을 통한 특성 선택

해양 어획량 분석 프로젝트

- **문제 (너무 낮은 성능)**

  실제 성능이 너무 낮은 문제를 보여줌, 어획량-> 어획률로 변경 하였지만 지속적인 낮은 성능을 보여줌



- **해결 방법**

1. 실제 어획량 및 어획률 데이터가 불안정해서 좋지 못한 성능을 보여 줌 -> 이유를 찾아보니 정책 및 인위적으로 어획에 변화를 주었음

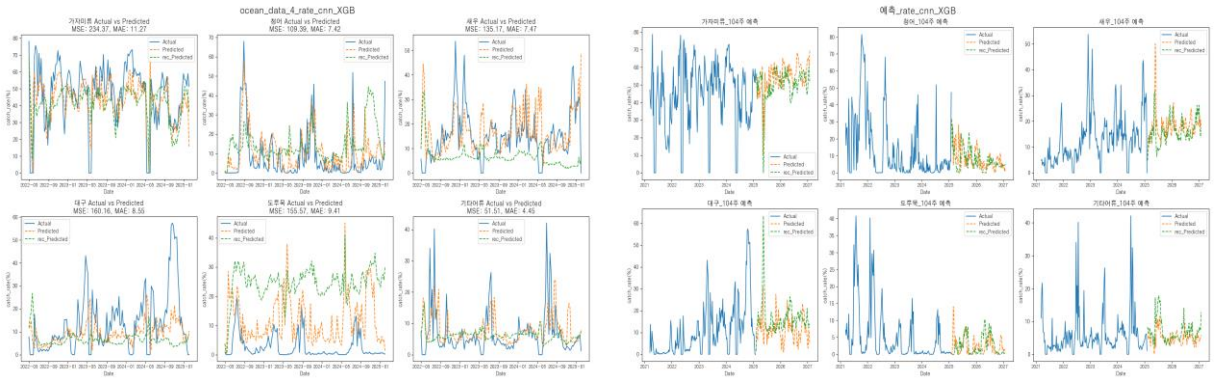
국내 최초 참가자미 대량 인공 부화…이달 방류

강원도, 겨울 **어종** '도루묵·뚝지' 치어 **방류**

KBS

네이버 뉴스 · 2025.04.03.

2. 논문을 통해 데이터수가 파라미터 수보다 적으면 과적합 및 훈련이 불가능 할 수 있다고 판단 -> ML모델(XGB,RF) 사용



- **최종 인사이트**

  해양 환경 변화가 실제 다양한 어류의 어획량 및 어획률에 큰 영향을 주지 못한다는 결론을 얻었습니다.

  더 나아가, 복잡하고 현실적인 데이터 구조와 인위적인 개입(조업 및 정책) 등 예측하기 어려운 비선형적인 요인들이 많아 기계학습 모델만으로는 어획량을 예측하는 데 한계를 가진다는 최종 결론을 도출했습니다.

표정 연습 애플리케이션 & 감정 기반 컨텐츠 추천

- **프로젝트 주제**

  표정 연습 애플리케이션 – 표정을 표현을 어려워 하는 사람들에게 재미와 연습을 동시에 제공 할 수 있는 앱을 제작

  감정 기반 컨텐츠 추천 – 감정에 따른 노래, 장문의 메시지, 대화를 통해 상대방의 감정을 케어 해주고 매일 해당 감정을 저장하여 유저의 감정 변화를 쉽게 파악

- **프로젝트의 어려움**

  한국인이라는 인종과 화남,기쁨,당황,슬픔이라는 적은 감정

- 프로젝트 일정 : 2025.06.16 ~ 2025.07.10
- 프로젝트 인원 : 5명
- 개인 기여도 : 60% (모델 학습,전처리,Flask,Flutter)
- 관련 스킬 : CV, CNN , Flutter , Flask , DB
- 세부스킬 : Yolo , postgresql , Pre-trained CNN

- 다양한 사전 학습 모델 사용

  1. ResNet50
  2. MobileNet v2 -> v3
  3. EfficentNetB0 -> B2 -> B3
  4. 최고 성능 + CBAM모듈
  5. VIT
  6. SWIN

모델	파라미터 수	연산량 (FLOPs)	F1-score	정확도
MobileNet	4.2M	0.3B	0.80	80.0%
EfficientNet	5.3M	0.39B	0.85	85.4%
CBAM + EfficientNet	5.8M	0.45B	0.85	85.3%
ResNet50	25.6M	4.1B	0.82	81.7%
SWIN	28M	4.5B	0.85	85.4%
ViT (Vision Transformer)	85M	55.4B	0.86	85.7%

표정 연습 애플리케이션 & 감정 기반 컨텐츠 추천

- **최종 모델 -> EfficentNetB0**

  경량화 모델이면서 성능이 가장 좋은 모델

- **문제점**

  성능이 가장 좋지만 일반화 성능이 너무 떨어짐
  처음에는 과적합 의심 파인튜닝을 단계적으로 진행 했음에도 지속적으로 같은 문제가 발생

- **해결 방법**

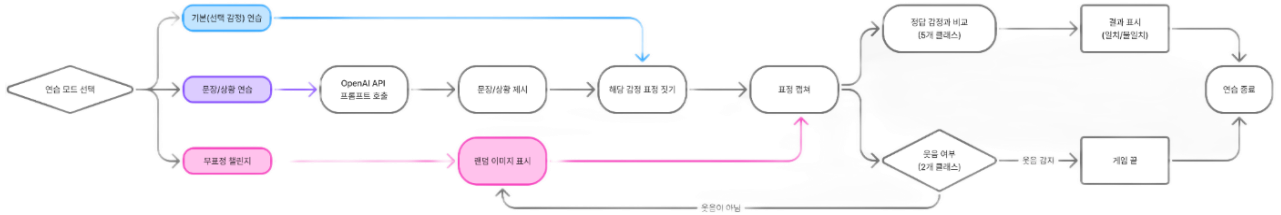
  B0은 조금의 복잡성이 있으면 제대로된 분류를 하지 못한다는 논문을 채택, B2로 재설정 후 검증 진행했더니 성능은 조금 좋아졌지만 아직까지도 불안정한 성능을 보여줌

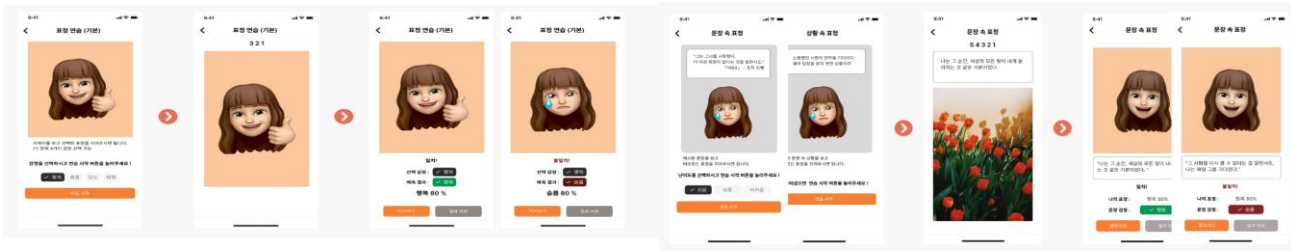
  최종적으로 B3 모델로 사용하여 모델 완성

- 모델 사용성

  감정 기반 컨텐츠 추천 -> 모델 기반으로 실시간 및 업로드한 이미지를 통해 노래, 장문의 메시지, 대화 를 추천 시스템 개발

표정 연습 애플리케이션





- **최종 결론**

  모델의 일반화 성능 검증의 중요성과 실제 환경 적용 시 발생할 수 있는 문제에 대한 깊이 있는 이해를 얻었습니다.

  특히, 다양한 인종과 다양한 표정, 그리고 다양한 각도의 이미지를 수집하여 더욱 정확한 성능의 모델을 개발할 수 있다는 중요한 인사이트를 얻었습니다

AI 기반 뉴스 요약 및 토픽 분석 서비스

- **프로젝트 개요**

  사용자가 원하는 뉴스를 실시간으로 수집하고 AI가 자동으로 요약하는 웹 서비스입니다. '제목-요약문' 형태의 직관적인 UI와 토픽 모델링, 워드클라우드 시각화를 통해 사용자는 복잡한 뉴스 속에서도 핵심 이슈와 트렌드를 빠르고 정확하게 파악할 수 있습니다.

- **사용 기술**
- Backend: Python, Flask, Flask-SocketIO
- Frontend: HTML, CSS, JavaScript
- NLP & Data Analysis:
  - Transformers (Hugging Face): 뉴스 요약
  - Mecab: 한국어 형태소 분석
  - Scikit-learn: 토픽 모델링 (LDA)
  - WordCloud: 데이터 시각화
  - Pandas: 데이터 처리
  - Crawling: Selenium, BeautifulSoup4

- **주요기능**
  - **실시간 뉴스 데이터 수집**: 사용자가 입력한 키워드와 기간에 맞춰 네이버 뉴스 데이터를 실시간으로 크롤링.
  - **AI 기반 자동 요약**: Transformers 라이브러리의 요약 모델을 활용하여 각 뉴스 기사의 핵심 내용을 자동으로 요약.
  - **토픽 모델링 및 분류**: Mecab과 Scikit-learn의 LDA 모델을 사용하여 수집된 뉴스들을 주요 주제별로 자동 분류.
  - **데이터 시각화**: 각 토픽별 핵심 키워드를 시각적으로 보여주는 워드클라우드 자동 생성.
  - **실시간 처리 과정 시각화**: 데이터 수집부터 분석까지의 전 과정을 프로그레스 바로 시각화하여 사용자에게 실시간 피드백 제공.

AI 기반 뉴스 요약 및 토픽 분석 서비스

- **사용 모델**

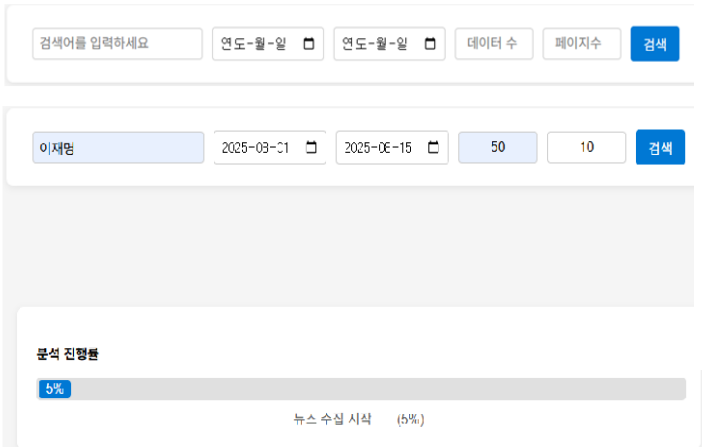
  허깅페이스 내 추출 모델을 통해서 크롤링 데이터를 내용을 추출 진행 후 eenzeenee/t5-base-korean-summarization에서 파인 튜닝을 진행

- **시도한 모델**
  - KoBert 기반 요약 모델
  - T5-small 모델

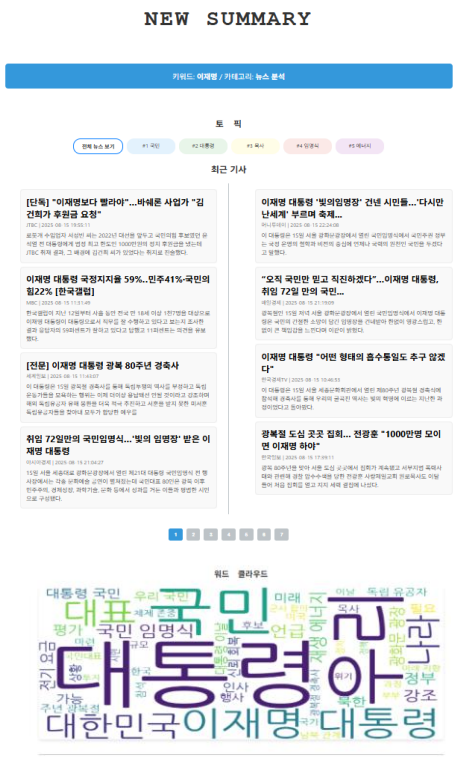
- **사용모델 선정 이유**

KoBERT는 긴 입력 토큰을 처리할 수 있는 장점이 있었으나, 실제 요약 결과물의 성능이 다소 불안정했습니다. 반면, T5 모델은 "요약:"과 같은 접두사(prefix)를 입력해야 하는 제약이 있었지만, KoBERT에 비해 **일관성 있고 높은 품질의 요약문**을 생성하여 최종 모델로 선정했습니다. 파인 튜닝을 통해 프로젝트 데이터에 특화된 성능을 확보하여 사용자에게 더 정확한 요약 정보를 제공할 수 있었습니다.

- 서비스 인터페이스



검색창.html



결과.html

- **기대효과 및 어려움**

많은 뉴스데이터를 AI를 통해 요약을 진행하여 유저들이 쉽게 뉴스 내용을 알고 그 뉴스를 접근 할 수 있게 도와줍니다. 하지만 프로젝트를 진행하면서 같은 내용의 뉴스가 3~4개씩 올리는 기자가 있을 만큼 중복내용도 존재 했기에 추후에는 유사도 검색을 통해 해당 뉴스들을 배제하는 방향으로 변환 예정입니다.

보이스 피싱 AI 탐지 서비스

- **프로젝트 개요**

  보이스피싱의 다양한 시나리오로 인해서 AI탐지 서비스가 있음에도 지속적으로 상승하고 있으며 최근 2030세대의 피해 비율이 증가하였습니다. 여기에 주요 요인은 젊은 세대들은 많은 정보를 알고 있기에 단순한 경고로는 해당 보이스피싱이 잘못 되었다고 판단하여 지속적인 연락을 한다고 판단하여 정보까지 제공 하는 서비스를 제작 하였습니다.

- 프로젝트 인원 : 4명
- 프로젝트 일정 : 2025.09.05 ~ 2025.09.25
- 개인 기여도 : 40% - Faiss 생성,BM25 생성, RAGAS를 통해 검색기 조율 및 청크단위 비교,전체적인 langgraph구조화

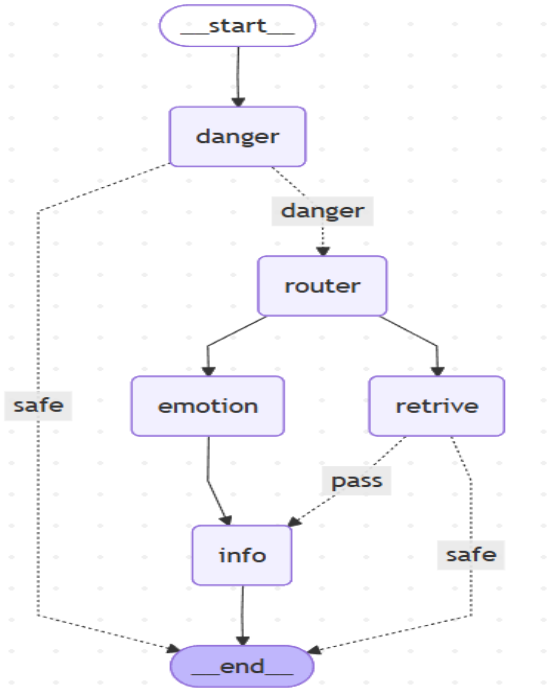
- **사용 기술**
- Backend: Python,FastAPI,Wepsocket
- Frontend: HTML, CSS, JavaScript
- Model: Rag,Langchain,Langgraph,Faiss,BM25,Cohere,openaiapi,

- **데이터 수집**
- 데이터는 금감원,경찰청에서 제공되어지는 피해사례 데이터를 사용하였으며, 음성 데이터 또한 실제 통화데이터를 사용하였습니다.
- **데이터 청크 사이즈**
- 아래 표 처럼 청크사이즈와 검색방식을 통해서 최종 방식을 선정

Model	Chunk Size	context_precision	context_recall	faithfulness	answer_relevancy
BM25	250	1.0	0.6905	0.7196	0.7667
BM25	500	1.0	0.9167	0.8091	0.7703
BM25	750	1.0	0.9167	0.7487	0.7675
BM25	1000	1.0	0.9167	0.8159	0.7602
Ensemble	250	1.0	0.9167	0.766	0.7765
Ensemble	500	1.0	0.9167	0.7556	0.7781
Ensemble	750	1.0	0.9167	0.7507	0.7746
Ensemble	1000	1.0	0.9167	0.9278	0.779
Retriever	250	1.0	0.9167	0.7569	0.772
Retriever	500	1.0	0.6905	0.6791	0.7805
Retriever	750	1.0	0.7738	0.5944	0.7848
Retriever	1000	1.0	0.9167	0.6009	0.7612

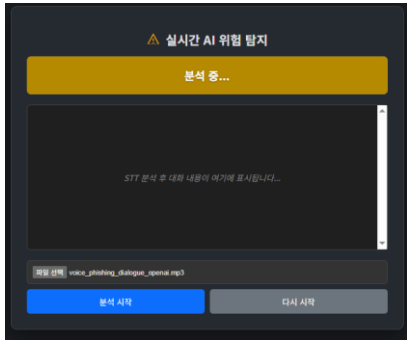
보이스 피싱 AI 탐지 서비스

- Langgraph 흐름
- 음성데이터가 보이스피싱이라고 의심이 되면 1차적으로 경고를 하면서 전체적인 분석을 들어갑니다. Emotion에는 심리를 통해서 어떻게 접근하는지를 작성해주며 RAG(retrive)에서는 실제 피해사례와 유사한 케이스를 가져옵니다. 이때 이전 사례와 최신 사례를 분류해서 가져옵니다. 검색을 후 cohere를 통해 rerank를 사용하여 보다 높은 관련성 데이터를 가져와서 마지막 Info에서 정보를 취합하여 보고서 형태로 출력이 됩니다.



- **고찰**
- 첫번째는 STT의 양방향 통신 구현 실패입니다. 아직까지는 어떻게 구성해야 할지를 감을 못잡아서 Tobe로 두었습니다.
- 두번째는 STT 모델의 어려움입니다. 처음 사용한 STT 모델은 whisper를 사용하였지만 생각보다 text가 제대로 나오지 않아 GCS도 사용하고 허깅페이스 내 모델도 사용하였지만 좋은 성능이 나오지 않았습니다. 그래서 한국어 특화 서비스를 제공하는 NCP Speech 모델을 사용하였으며 실제 성능은 가장 잘 나왔습니다. 하지만 이는 지속적인 비용이 발생하기에 추가 모델을 학습하여 하 한다는 점입니다

- **시연페이지**



음성데이터 입력

1차 경고


최종경고(info)