

김상겸 포트폴리오

검색 시스템과 데이터 파이프라인 구축 경험을 바탕으로, 효율적이고 확장 가능한 AI 서비스를 설계합니다. 벡터 검색, RAG 등 다양한 기술을 실제 서비스에 적용해왔으며, 새로운 기술을 통해 사용자 경험을 실질적으로 개선하는 일에 보람을 느낍니다.

Email: vhf1030@naver.com

Mobile: 010-9212-0252

GitHub:
github.com/vhf1030

구분	기술
언어	Python, R, Shell, Go
ML/DL	scikit-learn, PyTorch, CUDA, LightGBM, XGBoost, YOLO, LLaVA
LLM / RAG	LangChain, FAISS, ChromaDB, SBERT, E5, Gemini API, Ollama
데이터	Elasticsearch, MySQL, Filebeat, Logstash, Kibana
백엔드	Django, FastAPI, Streamlit, WebSocket
협업 / 운영	Git, Notion, Docker, Jupyter, Linux, Airflow, Ray

KOMI – 자세 분석 기반 재활 코칭 서비스

프로젝트 개요

- GitHub: <https://github.com/vhf1030/KOMI>
- 한 줄 설명: YOLO 기반 관절 분석과 의료 특화 LLM을 활용하여 원격 진단과 맞춤형 재활 운동을 실시간으로 제공하는 AI 재활 코칭 시스템
- 프로젝트 배경:
고령화와 운동 부족으로 인한 근골격계 질환 증가에 따라 병원 방문 없이 자가 운동을 할 수 있는 원격 재활 수요가 확대되고 있습니다. 본 프로젝트는 실시간 포즈 분석과 LLM 기반 피드백을 통해 보다 정확하고 지속가능한 재활 운동 환경을 제공하고자 합니다.
- 유형: 팀 프로젝트 (김상겸, 김형섭, 이장헌)
- 역할:
 - FastAPI 기반 서버 개발 및 WebSocket 통신 인프라 구축

- 포즈 인식 모델(YOLO) 및 LLM 응답 모듈 통합
- 클라이언트 WebCam 제어 모듈 및 스트리밍 처리 구현
- Streamlit 기반 UI 구성 및 실시간 분석 결과 시각화
- 진행 기간: 2025.03.10 ~ 2025.04.02
- 소속: 프로그래밍 기반 열라인 프로젝트 – 데이터/AI 트랙

기술 스택

- Python, FastAPI, Streamlit, OpenCV, YOLO, WebSocket, OpenAI

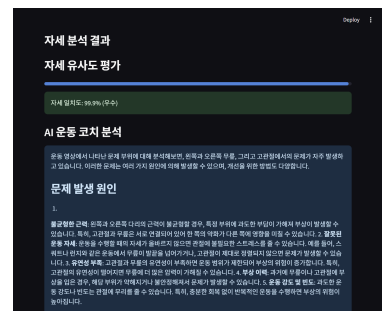
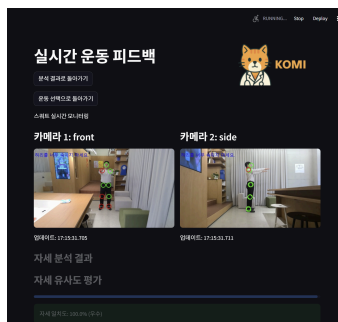
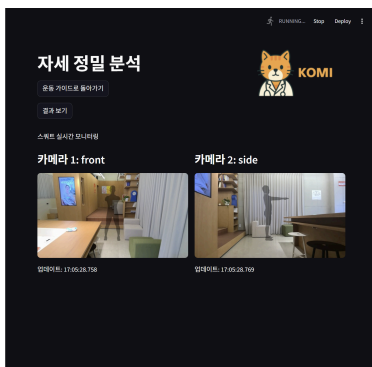
프로젝트 진행 단계

1. 문제 정의 및 서비스 요구사항 분석
2. 데이터 수집 및 처리 (운동 영상 및 텍스트 문서 기반)
3. 포즈 인식 모델 통합 및 실시간 추론 로직 구현
4. LLM 기반 피드백 생성 시스템 연동
5. 프론트엔드 및 UI 구성
6. 서버-클라이언트 통합 및 테스트

프로젝트 세부 과정

- 문제 정의
 - 실시간으로 자세 오류를 분석하고, 사용자에게 의료 기반 피드백을 제공하는 재활 서비스 구조 설계
 - 복잡한 설치 없이 간편하게 웹 기반으로 이용 가능한 환경 구성
- 서버 및 통신 인프라 구축
 - FastAPI 기반 API 서버 구축
 - 다양한 API 엔드포인트 구현: 상태 확인, 운동 종류 요청 등
 - WebSocket 기반 스트리밍 구조 설계 → 클라이언트에서 전송되는 이미지 프레임 처리
 - YOLO를 통한 관절 좌표 추출 → 비교/계산 → LLM으로 피드백 생성 및 전달
- 클라이언트 모듈 개발 및 통합
 - OpenCV로 WebCam 제어 및 프레임 캡처

- 포즈 분석 결과를 오버레이 처리하여 사용자에게 시각적으로 피드백 제공
- 분석 모드/가이드 모드/녹화 모드 등 다양한 모드 전환 기능 포함
- UI 및 시각화 구성
 - Streamlit 기반 인터페이스 설계
 - 운동 선택 → 분석 → 결과 제공 → 피드백 흐름으로 구성
 - 자세 점수, 문제 부위, 텍스트 피드백 등의 결과를 직관적으로 제공
 - 세션 상태 및 이벤트 루프 처리를 통해 사용자 간 충돌 방지



프로젝트 결과

- Streamlit + WebSocket 기반 MVP 완성
- 실시간 자세 인식 및 LLM 기반 피드백 서비스 연동 완료

Construction RAG - 건설사고 재발방지 대책 생성

프로젝트 개요

- GitHub: <https://github.com/vhf1030/construction-accident-prevention>
- 한 줄 설명: 건설 현장 사고 데이터를 기반으로 유사 사고에 대한 재발 방지 대책과 대응 방안을 자동 생성하는 로컬 기반 RAG 질의응답 시스템 개발
- 프로젝트 배경:

건설사고는 반복되는 유형이 많고 사고 발생 시마다 즉각적인 조치가 요구되지만, 상황에 따라 적절한 대응 방안을 수립하는 데 많은 시간과 전문성이 필요합니다. 본 프로젝트는 과거 사고 데이터를 활용하여 AI가 상황별로 재발 방지 대책과 향후 조치 방안을 자동으로 제시할 수 있도록 RAG구조 기반 시스템을 구축하는 것을 목표로 했습니다.

- 유형: 공모전 - 한솔데코 건설안전 시즌3 (김상겸, 김지민, 이장현, 이진규, 정재식)
- 역할:
 - 문서 기반 RAG 구조 설계 및 질의응답 시스템 개발
 - 문장 단위 검색 및 문맥 복원 로직 설계
 - 벡터 임베딩 및 검색 파이프라인 구축
 - LLM기반 응답 시스템 설계
 - 로컬 LLM 서빙 환경(Ollama) 설정 및 연동
 - 텍스트 기반 사고 분석(EDA) 및 프롬프트 설계
- 진행 기간: 2025.03.10 ~ 2025.03.28

기술 스택

- Python, LangChain, RAG, sbert, ChromaDB, Gemma-3b, Llama-8b, pandas

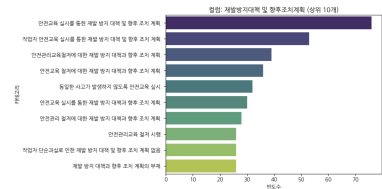
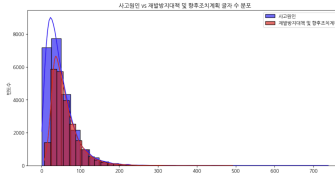
프로젝트 진행 단계

1. 문제 정의 및 사고 구조 분석
2. 데이터 수집 및 텍스트 전처리
3. 텍스트 기반 사고 유형 EDA
4. 문장 단위 분할 및 메타데이터 설계
5. 임베딩 생성 및 ChromaDB 저장
6. Retriever 및 RAG 질의응답 시스템 개발
7. LLM 모델 연동 및 테스트

프로젝트 세부 과정

- 데이터 수집 및 전처리
 - PDF는 문단 단위로 분리 후 문장 단위로 재구성하고, 각 문장에 문단 정보를 포함하는 메타데이터 부여

- CSV는 사고 유형, 사고 객체, 사고 원인 등 주요 항목을 활용해 질문-답변 형태로 전환
- 최종적으로 RAG 응답에 최적화된 형태로 데이터 가공
- 텍스트 기반 EDA
 - 사고 원인, 공정, 사고 객체 간 관계 분석
 - 자주 발생하는 사고 패턴과 사고-공종 매핑 파악
 - 분석 결과를 바탕으로 프롬프트 구성에 필요한 표현 방식 및 키워드 추출
- RAG 시스템 구성
 - 문장 단위 임베딩을 생성하고, 검색 시 문장 유사도를 기준으로 상위 문단을 복원하는 방식의 계층적 검색 구조 설계
 - 벡터 임베딩 결과는 ChromaDB에 저장
 - LangChain을 통해 검색된 문맥을 LLM 입력으로 활용하여 응답 생성
 - LLM은 Gemma-3b 및 Llama-VARCO-8B-Instruct를 선택적으로 연동하며, Ollama 기반 로컬 서버를 통해 인터넷 없이도 모델 추론 가능



프로젝트 결과

- 문장 단위 검색 기반 RAG 구조를 직접 설계하고, 실제 적용 가능한 응답 시스템으로 구현
- 로컬 LLM(Gemma-3b, Llama-VARCO)과의 안정적인 연동을 통해 외부 API 없이 독립 실행 가능
- 사고 데이터 기반의 프롬프트 템플릿 설계 및 QA 시스템으로 확장 가능성 확인

DeepDiary – 감정 일기 생성 서비스

프로젝트 개요

- GitHub: <https://github.com/vhf1030/DeepDiary>
- 한 줄 설명: 사용자의 사진과 대화를 기반으로 AI가 질문과 일기를 생성하고, 감정 분석을 통해 활동과 음악을 추천하는 대화형 일기 작성 지원 서비스
- 프로젝트 배경: 일기는 감정 정리와 스트레스 완화에 효과적이지만, 많은 사람들이 일기 작성을 어려워합니다. DeepDiary는 일기를 쉽게 작성하기 위해 사진과 대화를 입력으로 활용하고, AI가 감정을 분석하여 사용자가 자연스럽게 감정을 표현하고 콘텐츠를 생성할 수 있도록 유도합니다.
- 유형: 팀 프로젝트 (김상겸, 오인용, 이성복)
- 역할:
 - 전체 프로젝트 매니징 및 일정 관리
 - 이미지 캡셔닝 모델(LLaVA) 및 임베딩 모델(E5) 연동
 - FastAPI 서버 구현 및 AI 모델 서빙 구조 설계
 - Streamlit 기반 사용자 인터페이스 개발 및 입력 흐름 구성
- 진행 기간: 2025.02.20 ~ 2025.03.05
- 소속: 프로그래밍 기반 얼라인 프로젝트 – 데이터/AI 트랙

기술 스택

- Python, FastAPI, Streamlit, pandas, PyTorch, LLaVA, E5, Gemini API

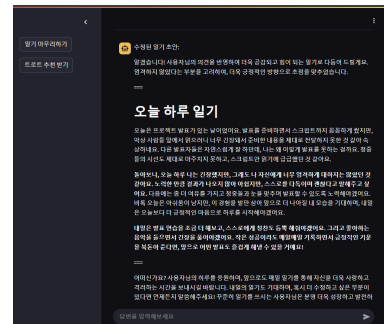
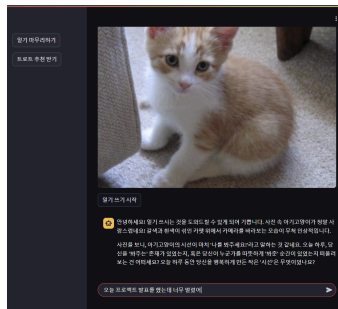
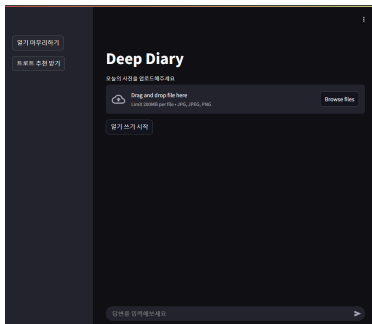
프로젝트 진행 단계

1. 문제 정의 및 기획
2. 데이터 수집 및 전처리
3. 멀티모달 AI 모델 개발
4. API 및 백엔드 구축
5. 프론트엔드 구현 및 서비스 연동

프로젝트 세부 과정

- 문제 정의 및 기획
 - 사람들이 일기 작성을 어렵게 느끼는 원인을 분석
 - 이미지 기반 입력과 자연스러운 대화를 결합하면 진입 장벽을 낮출 수 있다는 점에 착안

- 감정 분석과 콘텐츠 추천을 연동한 구조 설계
- AI 모델 개발
 - LLaVA 기반 이미지 캡셔닝 모델 연동: 사진을 입력받아 설명 문장을 생성
 - E5 임베딩 모델 적용: 감정 벡터를 통해 유사한 트로트 가사 추천
 - Gemini API 사용: 사용자 입력 기반 질문 유도 및 일기 요약, 감정 분석
- API 및 백엔드 개발
 - FastAPI 서버 구축
 - AI 모델 호출 및 결과 리턴 구조 설계
 - 사용자 요청 흐름에 맞춘 엔드포인트 설계
- 프론트엔드 개발
 - Streamlit 기반 웹 UI 구성
 - 사진 업로드 → 질문 응답 → 일기 생성 → 감정 분석 → 음악 추천까지의 흐름 설계



프로젝트 결과

- FastAPI + Streamlit 구조로 완성된 AI 기반 감정 일기 작성 도우미 MVP 구현
- 이미지와 대화를 결합한 자연스러운 일기 작성 경험을 제공

Pisces - 수산시장 가격 예측

프로젝트 개요

- GitHub: <https://github.com/vhf1030/Pisces>

- 한 줄 설명: 머신러닝을 활용한 시장별 어종별 수산물 가격 예측 시스템
- 프로젝트 배경: 수산물 시장은 계절, 기상, 환율, 수요 변동 등 다양한 외부 요인에 따라 가격 변동성이 매우 큼. 본 프로젝트는 머신러닝을 활용한 가격 예측 모델을 통해 향후 가격 흐름을 예측함으로써, 소비자와 유통업자에게 보다 합리적인 구매·유통 의사결정을 지원합니다.
- 유형: 팀 프로젝트 (김상겸, 원정환, 지서연)
- 역할:
 - 다양한 출처의 데이터 수집 및 통합
 - 결측치 처리, 원-핫 인코딩 등 전처리 수행
 - LightGBM 기반 가격 예측 모델 설계 및 튜닝
 - Django 기반 API 구축 및 서비스 연동
- 진행 기간: 2025.01.03 ~ 2025.02.04
- 소속: 프로그래밍 기반 얼라인 프로젝트 – 데이터/AI 트랙

기술 스택

- Python, pandas, scikit-learn, LightGBM, XGBoost, Django

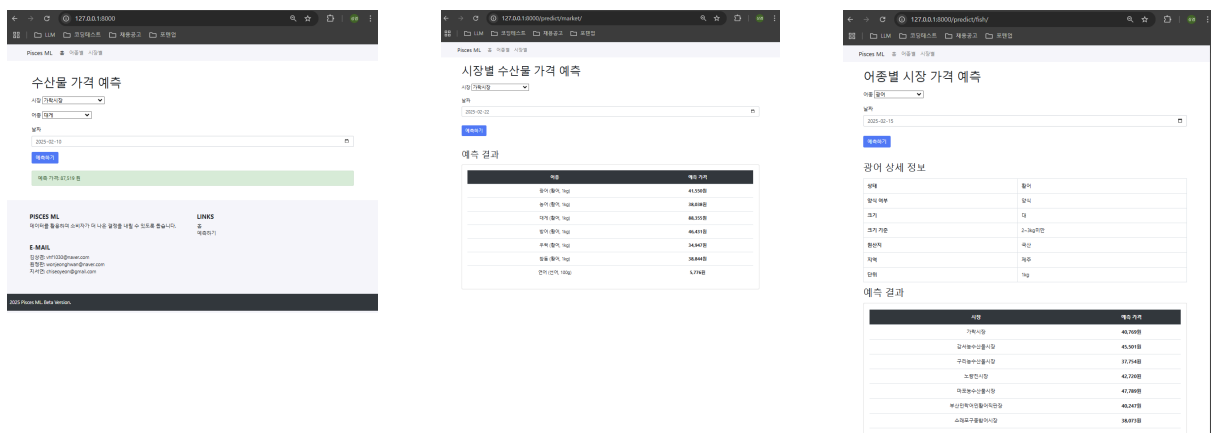
프로젝트 진행 단계

1. 문제 정의 및 데이터 수집
2. 데이터 전처리 및 탐색적 분석
3. 모델 개발 및 성능 검증
4. 인사이트 도출
5. 웹 서비스 구축

프로젝트 세부 과정

- 데이터 수집:
 - 수산물 가격(인어교주해적단), 기상 데이터(기상청), 환율(야후파이낸스), 검색 트렌드(네이버랩)
 - 일자별, 시장별, 어종별 가격 정보를 수집하여 일 단위 기준으로 통합
- 데이터 전처리 및 분석:
 - 결측치는 선형 보간법 및 평균 대체를 통해 보완

- 범주형 변수(시장명, 어종 등)는 원-핫 인코딩 처리
- 기온, 환율 등의 외부 변수와 가격 변동 간의 상관관계를 분석
- Lag Feature 및 시계열 특성 반영
- 모델 개발 및 평가:
 - Random Forest, XGBoost, LightGBM 등을 비교 실험
 - AutoML을 통해 LightGBM 모델의 하이퍼파라미터 튜닝
 - RMSE, R^2 를 기준으로 모델 성능 비교
 - 최종적으로 LightGBM 모델을 채택하여 시장별 예측
- 서비스 구현:
 - Django를 이용해 웹 API 서버를 구축하고 모델 연동
 - 사용자가 시장 및 어종을 선택하면 해당 날짜의 예측 결과를 응답하는 엔드포인트 구현



프로젝트 결과

- 학습 모델을 활용한 가격 예측 결과를 실제 API 형태로 서비스 연동
- MVP 수준의 웹 애플리케이션 구축 완료