

# 김민규



## 학력 사항

2017.03 - 2023.09 부산대학교 산업공학과 학사 / 통계학 부전공

2023.09 - 2024.03 모두의 연구소 부트캠프, AI 엔지니어 양성과정

## 경력 사항

2024.04.01 - 현재 Visuworks Co. | Data Scientist

요약 - 스타트업에서 데이터과학자로 일하며 다음과 같은 경험들을 했습니다.

- 3개의 AI/ML 프로젝트를 주도하며 개발부터 배포 및 운영까지 전 과정을 설계하고 구현했고, 이 과정에서 전체 파이프라인에 대해 깊게 이해할 수 있었습니다.
- 서비스를 운영하며 사용자의 피드백을 수집하고 개선해나가면서 만족도를 높여나가는 경험을 했고, 단순히 기술구현이 아닌 실제 비즈니스에 영향을 주기 위해 노력했습니다.
- 단순히 시키는대로 개발하는게 아닌, 문제의 본질을 생각하고 새롭게 정의하며 문제를 해결하는 본질적인 해결책을 제시했습니다.

## 주요 프로젝트

### 렌즈 사이즈 추천 서비스

2025.03 - 현재 (6개월)

- 성과/결과

인과추론 기반 예측 모델 개발 및 ML 파이프라인 구축을 통해 의료진의 신뢰도 향상 및 안정적인 ML 서비스 운영

- 해결과정

- 평가 문제: 선택되지 않은 선택의 예측결과를 어떻게 평가할 것인가
  - causal inference의 common support assumption을 통해 이론적 근거 확보
  - propensity score를 추정해 신뢰할 수 있는 범위 추정
  - partial identification 또는 causal model을 통한 robustness 확보
- 데이터 일관성 확보: 학습환경과 추론 환경의 불일치로 인한 data shift 문제
  - 데이터 파이프라인 표준화와 피쳐 스토어 기반의 피쳐 일관성 관리
- 모니터링 시스템 개발: 운영 환경에서 모델의 예측 성능을 실시간으로 확인하며, 외부에서 바로 드러나지 않는 silent failure를 조기에 탐지
- 도메인 특화 모델링: 임상의 직관과 일치하도록 임상적으로 의미 있는 피쳐 엔지니어링과 사용자(의사)의 직관에 맞는 모델링
  - monotonicity constraint을 적용하고, Shapley value 기반의 설명을 제공하여 의료진의 신뢰도를 확보

## 고객상담용 챗봇

2024.11 - 2025.03 (4 months)

### • 성과/결과

일평균 100~200건의 상담을 자동화하여 상담사 업무 효율성을 향상시키고, 분류 모델 도입으로 사용자 불만족도를 50% 감소시킨 RAG 기반 챗봇 시스템 구축

### • 해결과정

#### 1. 불만족도도 감소를 통한 사용자 경험 개선

- 사용자 피드백을 통해 불만족이 발생하는 부분 발견 : 대답할 수 없는 질문의 경우 답이 되지 않는 응답을 반복
- 분류 모델을 통해 질문을 분류하고, 종류별로 적절한 파이프라인 적용
- A/B test를 통해 만족도를 높일 수 있는 답변 생성 프롬프트 작성

#### 2. 검색 성능 향상

- Semantic search의 한계 : 사전 학습에 포함되지 않은 내용에 대한 검색 한계와 함께, 방법론 자체의 한계
- keyword filtering을 통해 도메인에 특수한 단어에 대한 검색 성능 향상
- 하이브리드 검색 도입해 semantic search의 한계 극복

#### 3. 대화 맥락에 맞는 답변 생성

- 질문을 재작성해 흐름에 맞는 문서를 검색하고 답변 생성

#### 4. 모니터링 시스템 : 모니터링 자동화를 통해 유지보수 비용 최소화

- RAGAS framework를 참고해 답변 생성이 문서에 기반했는지를 측정하는 faithfulness score를 통해 모니터링 자동화

## OCR 파이프라인

2024.07 - 2024.10 (4 months)

### • 성과/결과

객체 지향 설계와 단위 테스트를 통해 오류율 1% 이하, 문제에 특화된 OCR 모델 도입으로 OCR 정확도 99% 이상 달성

### • 해결과정

#### 1. 기존의 코드의 품질 개선 : 절차지향적이고 구분이 없어 디버깅이 힘들었던 기존 코드 개선

- 객체지향 설계 원칙 적용으로 에러 범위와 원인 파악이 명확해져 디버깅 효율성 80% 향상
- 단일 책임 원칙과 추상화를 통한 모듈화로 코드 가독성 향상 및 유지보수성 개선

#### 2. OCR 처리 속도를 평균 87% 가까이 향상시켜 사용자 만족도 향상

- 비동기 구조 도입으로 OCR 처리 시간과 Client 응답 시간을 분리하여 전체 시스템 처리량을 향상

#### 3. CI/CD 파이프라인 구축: 검증 프로세스를 통한 런타임 에러 감소

- pytest를 통한 unit test와 mypy를 활용한 타입 검증을 CI 과정에 통합하여 런타임 에러를 사전에 방지하고, 코드 품질을 지속적으로 관리할 수 있는 자동화된 개발 프로세스

#### 4. 모델 최적화 및 배포: 문제에 특화된 모델 사용해 정확도 99% 달성

- 검사장비의 고정된 이미지 구조를 활용해 Text Detection 단계를 생략하고 TrOCR 모델을 사용한 Text Recognition에 특화된 솔루션을 구현
- TritonServer를 활용한 동적 배치 처리로 처리량을 최적화