Data Scientist/ Machine Learning Engineer Portfolio

Data Science, Data Analysis, Data Engineering, Project Design, Project Management

Kwangmin Kim

2025-08-05

[P6] 자연어 처리(NLP)를 활용한 Data Governance 시스템 단계적 구축

프로젝트 개요

- 소속: Seegene
- 기간
  - Phase1: 데이터 표준화 시스템 구축 2024.10 ~ 2025.08 (진행중)
  - Phase2~3: 데이터 거버넌스 시스템 구축 2025.09 ~ 2027.09
- 참여인원: 20명 (Data Scientist, Data Engineer, SW 개발자, BT 개발자, DBA)
- 총괄장 수명 프로젝트
- 대내적 의의: 전사 자동화 시스템 구축의 시발점
  - 실험 자동화, 시약 개발 자동화, 분석 자동화, Data Driven 의사 결정
- 대외적 의의: 글로벌 기술 공유 사업의 시발점
  - Microsoft, Springer Nature, KPMG 등 각 국 정부기관 및 기업 등
- 역할: Technical Lead
  - 표준화 체계, 아키텍처 및 프로세스 구축
  - 1명의 Junior Data Scientist 멘토링: 문제정의, 데이터 분석 역량 강화
  - 19명의 IT/BT 개발자 멘토링: 데이터 거버넌스 70% 이해도 달성

주요 문제점 및 도전과제

- **문제점**
  - 16개 부서 53개 DB의 83% 메타데이터 불일치로 인한 데이터 활용도 저하 문제
  - 데이터 거버넌스 체계 부재, Data Silo 현상, 데이터 통합 및 검증 체계 부재
- **도전과제**
  - 표준화 체계 구축
    - 문제: 독립적으로 개발된 시스템 및 외주 개발 시스템 통합 불가
    - 해결방안: 표준화 현황 분석 및 표준화 프레임워크 확립
  - 데이터의 품질 평가 자동화
    - 문제: 영문 약어 생성 규칙 구현의 어려움 & 표준화 KPI (품질 평가 지표) 부재
    - 해결방안: 계층적 Rule Engine 설계 & 원칙 기반 평가 지표 개발
  - BT & IT 용어 표준화
    - 문제: (관용어 vs 표준화 원칙) & (SI 단위계 vs 업계 관행)
    - 해결방안: 업무 전문가와의 협업을 통한 사전 구축
  - 조직내 낮은 Data 성숙도 및 교육의 어려움
    - 문제: 임원진과 실무진의 막연한 두려움 & 표준화 지식 부족
    - 해결방안: 추상적 개념 구체화, 동기부여 및 교육

솔루션 설계 및 전략

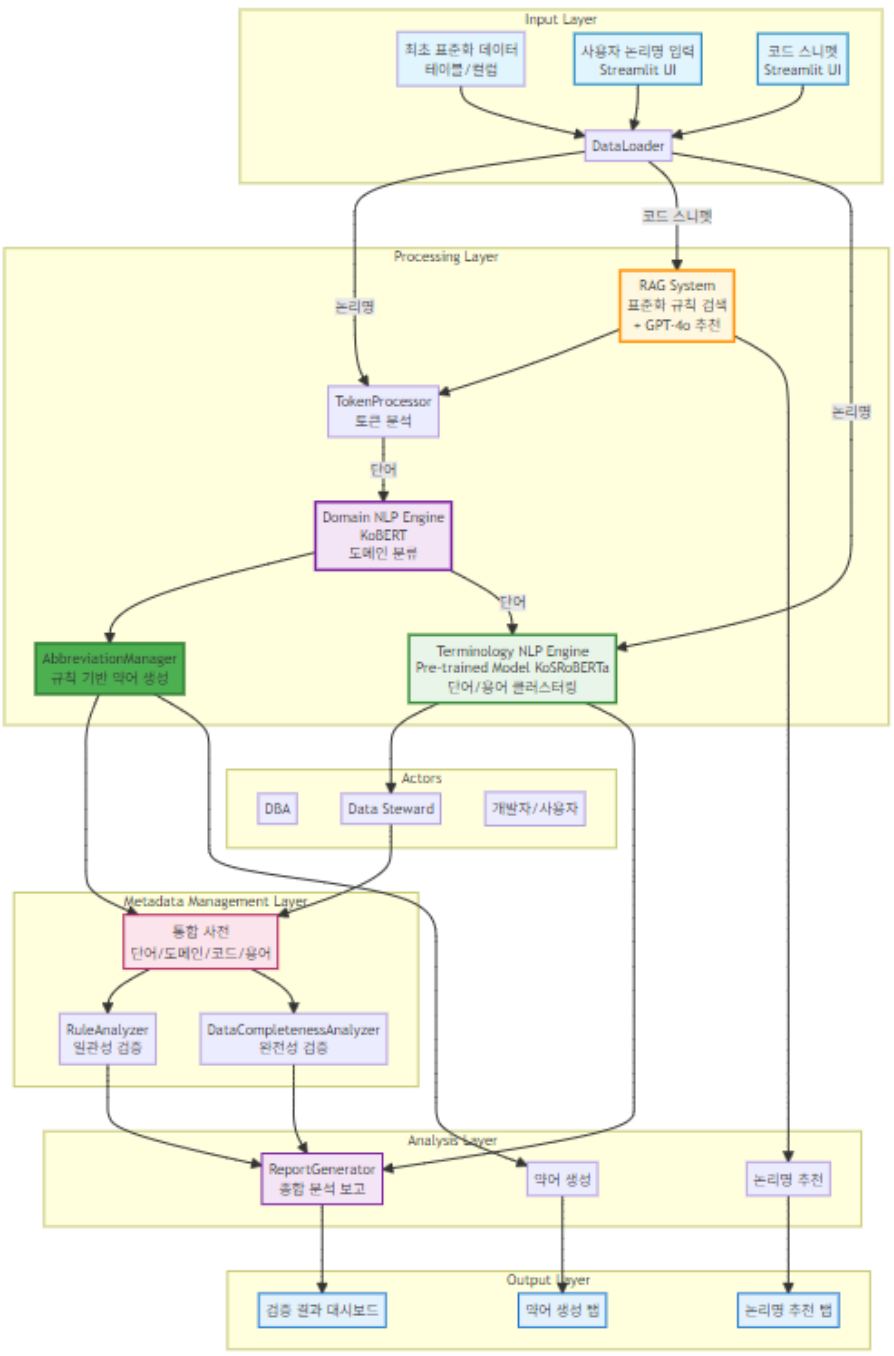

Figure 1: Data Standardization Architecture

- Phase 1: 표준화 프레임워크 구축 (Proof of Concept)
  - 표준화 프래임워크 실증 + 현실적 범위 설정 + 단계적 표준화 진행
  - 표준화 프레임워크 구축 (원칙 + 표준 사전 + 품질 평가)
  - 표준화 프로세스 반자동화 시스템 구축: Hybrid Rule Engine
    - Rule Based 표준화 프로세스 구축
    - Data Driven (딥러닝을 활용) 표준화 프로세스 병목 현상 해결
- Phase 2 ~ 3: 표준화 범위 확장 및 데이터 거버넌스 시스템 구축
  - DB 확장 시 표준화 프로세스 시스템 확장
  - MS Azure Data Factory & Databricks 연동
  - 거버넌스 체계 구축, 표준화 교육 및 캠페인 진행
  - 표준화 모니터링 시스템 구축
  - 자동화 워크 플로우 구축
  - 데이터 활용방안 모색

기술 스택 및 요구 역량

- 데이터 처리: Python (pandas, numpy, regex, NLTK, konlpy, pyarrow)
- 자연어 처리: Pytorch (Hugging Face, Transformer, KoBERT, KoSRoBERTa)
- 머신러닝: scikit-learn (HDBSCAN, PCA, UMAP, silhouette score, 등)
- 시각화 & 모니터링: Streamlit, matplotlib/seaborn, plotly, NetworkX
- RAG: Langchain, OpenAI (GPT-4o), FAISS, Hugging Face, pdfplumber
- 기술적 역량
  - 복잡한 비즈니스 규칙의 알고리즘 설계 능력
  - 딥러닝 모델 설계 및 훈련 데이터 생성 능력
- 업무적 역량
  - 데이터 표준화 프레임워크 구축 (표준화 원칙 및 표준 사전 구축)
  - 도메인 전문가와의 협업 및 요구사항 분석
  - 표준화 정책 수립 및 이행 관리
  - 경영진 및 실무진과의 커뮤니케이션 능력

결과 및 성과

- 정량적 성과: 데이터 품질 향상
  - 표준화 체계 수립: 0% → 100% (최초 구축)
    - 표준화 원칙, 표준 사전, 표준 데이터 계층 구조, 품질 평가 프로세스 구축
  - 메타데이터 완전성: 29.6% → 100% (80.4% 개선)
  - 메타데이터 일관성: 8.4% → 98.7% (90.3% 개선)
- 효율성 개선
  - 물리명 규칙 검증 시간: 수동 4시간 → 자동 0.73초 (99% 단축)
  - 약어 생성 정확도: 수동 생성 63% → 자동 100% (37% 향상)
  - 딥러닝을 활용한 표준화 프로세스 간소화: 도메인 분류 자동화
- 시스템 구축 성과
  - 표준화 (품질 평가 및 약어생성) 프로그램 핵심 모듈 8개 개발
  - 표준화 세부 규칙 200여개 생성
  - 품질 지표 16개 개발 및 자동 산출 체계 구축
  - 자연어 처리를 활용한 용어별 도메인 그룹 분류 및 유사 용어 그룹핑
    - 도메인 그룹 분류: 용어 중복 방지, 도메인 항목 관리 및 도메인 그룹 관리 (예시: "USER_ID", "고객번호", "제품코드", "비밀번호" 등)
    - 총 5,632개 훈련 용어, 14개 도메인 그룹, 훈련/테스트 분할 80%/20%
    - 유사 용어 Clustering: 금칙어 관리 및 표준안 관리 (예시: "사용자ID", "UserID", "User", "user_id" 등 금칙어 관리)
    - monologg/kobert (한글 분류 정확도 95.65%), jhgan/ko-sroberta-multitask (Clustering Silhouette Score, 0.2752 -> 0.5374)
  - RAG(LangChain + FAISS + OpenAI API)을 활용한 표준화 지원
    - 표준화 원칙 정보 검색 및 QnA
    - 코드 스니펫을 입력받아 논리명 추천
    - 데이터 표준화 원칙 (약 200개 Rules) 기반 논리명 추천
    - LLM 논리명을 입력받아 커스텀 약어생성모듈로 물리명 추천
    - 표준화 관련 회의 및 QnA 건 수 감소 (약 70건/주 → 약 4건/주, 94.3% 단축)
- 정성적 성과
  - 조직 차원: 16개 부서 통합 데이터 표준 확립
  - 표준화 정책 수립 및 거버넌스 체계 구축

기대효과

- 단기 기대효과
  - 데이터 통합 작업 시간 단축 (2시간 → 약 5분, 95.8% 단축)
  - 신규 시스템 구축 시 신속한 표준안 적용 가능
  - 데이터 품질 이슈 사전 예방 체계 확립
- 장기적 비즈니스 가치
  - 부서 간 데이터 공유 활성화, 통합 작업 시간 단축 및 활용도 향상
  - 실험 데이터 모니터링 시스템 구축
  - 글로벌 기술 공유사업 본격화

추후 과제

- 표준화 범위 확대
- Airflow를 활용한 자동화 워크플로우 구축
- 표준 사전 및 DB 메타데이터 데이터베이스 연동
- 모니터링 및 성능 최적화

[P6-Rule Engine] 표준화 규칙에 기반한 표준화 품질 평가 시스템

프로젝트 개요

- **컨텍스트**: 본 프로젝트는 "자연어 처리(NLP)를 활용한 Data Governance 시스템"의 하위 프로젝트에 해당하며, 전체 시스템의 핵심 기반이 되는 표준화 규칙 엔진을 구축하는 단계
- **연관 프로젝트**: [P6-Rule Engine] → [P6-NLP Engine] → [P6-RAG Engine] → [P6-MLOps]
- 기간: 2024.10 ~ 2025.08 (진행중, 전체 프로젝트의 핵심 모듈)
- 참여인원: Data Scientist 1명 (단독 설계 및 구현)
- 전사 데이터 표준화 프로젝트의 핵심 엔진 개발
  - 16개 부서 53개 DB의 83% 메타데이터 불일치 문제 해결을 위한 품질 평가 엔진
  - 복잡한 한국어/영문 명명 규칙을 체계적으로 검증하는 자동화 시스템 구축
  - 실무자들의 약어 생성 병목 현상 해결을 위한 지능형 약어 생성기 개발
- 역할: Technical Lead & 알고리즘 설계자
  - 16단계 명명 규칙 체계 설계 및 구현
  - 복잡한 비즈니스 규칙의 알고리즘화 (200여개 세부 규칙 중 부분 구현)
  - 약어 생성 엔진 및 중복 해결 전략 설계
  - 품질 평가 지표 16+개 개발 및 자동 산출 체계 구축

주요 문제점 및 도전과제

- **기술적 문제점**
  - 표준화 담당자 (data stewards) 4명이 단기간내 표준화 규칙 완벽 숙지 어려움
  - 표준화 원칙에 대한 소형 ChatBot 및 **표준화 작업 및 평가 자동화 시스템 구축** 필요
  - 품질 평가를 표준화 담당자 육안과 수기로 진행하는 것은 사실상 불가능하다는 것을 확인
  - 복잡한 약어 생성(=물리명) 규칙의 알고리즘화 어려움
    - 문제: 자음/모음 분류, 연속 자음 처리, 모음 삽입 위치 결정 등 언어학적 복잡성
    - 해결방안: 토큰 프로세서를 통한 문자 단위 분석 및 규칙 기반 처리
  - 약어 생성 시 중복 해결의 복잡성
    - 문제: 동일한 약어가 생성되는 경우의 체계적 해결 방안 부재
    - 해결방안: 3단계 중복 해결 전략 (남은 자음 추가 → 모음 위치 기반 추가 → 순번 부여)
  - 다양한 예외 상황 처리
    - 문제: 통용 약어, 접두사, 4글자 이하 단어 등 특수 케이스 대응
    - 해결방안: 전략 패턴을 활용한 다층 처리 구조 설계
- **운영적 도전과제**
  - 실시간 대용량 데이터 처리 요구사항
    - 문제: 수천 개 메타데이터의 실시간 품질 평가 필요
    - 해결방안: 효율적인 알고리즘 설계로 4시간 → 0.73초 달성
  - 복잡한 규칙의 유지보수성 확보
    - 문제: 200여개 세부 규칙의 변경 시 시스템 안정성 보장
    - 해결방안: 모듈화된 규칙 검증기(RuleChecker) 패턴 적용
  - 다양한 사용자 요구사항 대응
    - 문제: DBA, Data Steward, 개발자 등 서로 다른 사용자 그룹의 요구사항
    - 해결방안: 품질 평가 통합 보고서 및 약어 생성 위반 추적 기능 제공
- **제약 조건**
  - 기존 시스템과의 호환성 유지 필요
  - 높은 정확도 요구 (90% 이상)
  - 확장 가능한 아키텍처 설계 필요

솔루션 설계 및 전략

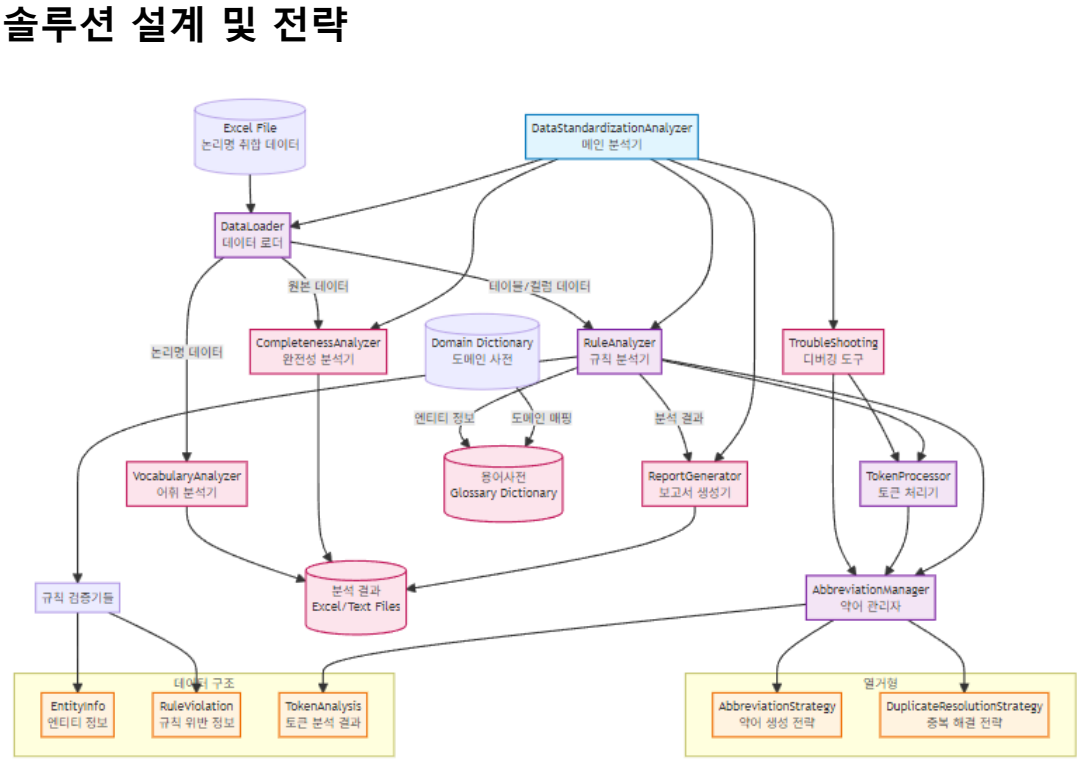

Figure 2: 표준화 품질 평가 시스템 아키텍처

- **1단계: 16단계 명명 규칙 체계 설계**
  - 전처리 규칙 (1-4단계): 불용어 제거, 통용 약어 처리, 접두사 분리, 짧은 단어 처리
  - 문자 분류 규칙 (5-7단계): 모음/자음 구분, Y/W 자음 처리, 첫 글자 모음 자음화
  - 약어 생성 로직 (8-10단계): 자음 우선 구성, 연속 자음 처리, 모음 삽입
  - 후처리 및 검증 (11-14단계): 중복 처리, 유일성 보장
  - 표현 규칙 (15-16단계): 케이싱 처리, 길이 제한
- **2단계: 계층적 아키텍처 구축**
  - TokenProcessor: 한국어/영문 논리명의 토큰화 및 문자 분석
  - AbbreviationManager: 다전략 약어 생성 및 중복 해결
  - RuleAnalyzer: 규칙별 위반 검증 및 상세 분석
  - ReportGenerator: 시스템별/유형별 종합 품질 리포트 생성
- **3단계: 지능형 약어 생성 전략**
  - 기존 약어 재사용 → 통용 약어 적용 → 원본 유지 → 신규 생성 순으로 우선순위 적용
  - 중복 발생 시: 남은 자음 추가 → 모음 위치 기반 추가 → 순차 번호 부여
  - 핵심 처리 규칙:
    - 자음/모음 분류: Y, W를 자음으로 처리 (규칙 6)
    - 첫 글자 특수 처리: 단어 첫 글자가 모음인 경우 자음으로 취급 (규칙 7)
    - 연속 자음 압축: "buffer" → "bfr" (같은 자음 연속 시 두 번째 제거, 규칙 9a)
    - 모음 삽입 로직: 자음 4글자 미만 시 원본 위치 기준 모음 보강 (규칙 9b, 10)
- **4단계: 품질 평가 지표 개발**
  - 일관성 평가: 물리명이 표준화 규칙 기반 생성되었는지 확인
  - 완전성 평가: 필수 메타데이터 누락 여부 확인
  - 유사 용어 클러스터링: jhgan/ko-sroberta-multitask 활용
    - 목적: 금칙어 관리 및 표준안 관리 ("사용자ID" vs "UserID" 통합)
    - 성과: DBA와 협업하여 테이블 정규화 후 Silhouette Score 0.2752 → 0.4374 (66.2% 개선)
  - 도메인 그룹 분류: 용어 중복 방지 및 도메인별 관리
  - 표준화 대상 우선순위: 클러스터 크기 기반 표준화 우선도 결정

기술 스택 및 요구 역량

- **핵심 기술**
  - 언어: Python (pandas, numpy, regex, dataclasses)
  - 아키텍처 패턴: Strategy Pattern, Factory Pattern, Observer Pattern
  - 알고리즘: 문자열 처리, 그래프 탐색, 동적 프로그래밍
  - 데이터 처리: pandas, 대용량 Excel 파일 처리, 메모리 최적화
  - 테스팅: pytest, 단위 테스트, 통합 테스트, 성능 테스트
- **기술적 역량**
  - 알고리즘 설계: 복잡한 비즈니스 규칙의 효율적 알고리즘 변환
  - 소프트웨어 아키텍처: 확장 가능하고 유지보수 용이한 모듈 설계
  - 문자열 처리: 한영 혼재 텍스트의 토큰화 및 명명 규칙 구현
- **업무적 역량**
  - 요구사항 분석: 도메인 전문가와의 협업을 통한 관용어 및 금칙어 목록 협의
  - 품질 관리: 테스트 주도 개발 및 지속적 품질 검증
  - 문서화: 200+개 규칙과 표준 사전들에 대한 체계적 문서화
  - 사용자 교육: 다양한 사용자 그룹 대상 시스템 활용 교육

결과 및 성과

- **정량적 성과: 극적인 성능 향상**
  - 물리명 규칙 검증 시간: 수동 8시간 → 자동 0.73초 (99.98% 단축)
  - 약어 생성 정확도: 수동 생성 63% → 자동 100% (37% 향상)
  - 메타데이터 일관성: 8.4% → 98.7% (90.3% 개선)
  - 메타데이터 완전성: 29.6% → 100% (80.4% 개선)
- **시스템 구축 성과**
  - 핵심 모듈: 표준화 품질 평가 핵심 모듈 9개 개발 완료
  - 세부 규칙: 200+개 표준화 세부 규칙 알고리즘화 완료
  - 평가 지표: 16개 품질 지표 개발 및 자동 산출 체계 구축
  - 아키텍처: 확장 가능한 모듈형 아키텍처로 향후 규칙 추가 용이
- **기술적 성과**
  - 자동화 달성: 16단계 명명 규칙의 완전 자동 검증 시스템 구축
  - 정확도 향상: 수동 대비 약어 생성 정확도 100% 달성
  - 모듈화 설계: TokenProcessor, AbbreviationManager 등 9개 핵심 모듈 구축
  - 성능 최적화: set 자료구조 활용한 중복 검사 및 배치 처리 최적화
- **업무 효율성 개선**
  - 자동화 달성: 기존 수동 검증 프로세스 100% 자동화
  - 품질 일관성: 사람에 따른 검증 결과 편차 제거
  - 실시간 처리: 대용량 메타데이터 실시간 품질 평가 가능
  - 확장성 확보: 새로운 규칙 추가 시 기존 시스템 영향 최소화
- **조직적 임팩트**
  - 표준화 문의 감소: 약 70건/주 → 약 4건/주 (94.3% 감소)
  - DBA 업무 효율: 메타데이터 검증 업무 시간 95% 단축
  - 데이터 거버넌스: 전사 데이터 품질 모니터링 체계 구축 기반 마련
  - 지식 체계화: 암묵적 명명 규칙의 명시적 알고리즘화 완료

기대효과

- **단기 기대효과**
  - 신규 시스템 구축 시 실시간 표준 검증 가능
  - 데이터 통합 프로젝트 시 품질 이슈 사전 예방
  - 메타데이터 관리 업무 효율성 지속적 향상
- **장기적 비즈니스 가치**
  - 전사 데이터 거버넌스 체계 수립의 기반 마련
  - 데이터 품질 기반의 신뢰성 있는 의사결정 지원
  - 시스템 확장 시 표준화 체계 재사용 가능

[P6-NLP Engine] KoBERT 기반 한국어 도메인 분류 시스템

프로젝트 개요

- **컨텍스트**: 본 프로젝트는 "자연어 처리(NLP)를 활용한 Data Governance 시스템"의 용어의 의미를 파악하여 도메인 그룹을 자동 분류함으로써 실무자들의 병목 현상을 해결
- **연관 프로젝트**: [P6-Rule Engine] → **[P6-NLP Engine]** → [P6-RAG Engine] → [P6-MLOps]
- 기간: 2024.12 ~ 2025.03 (4개월, [P6-Rule Engine] 완료 후 진행)
- 참여인원: Data Scientist 1명 (단독 설계 및 구현)
- 전사 데이터 표준화 프로젝트의 NLP 엔진 개발
  - 5,632개 훈련 용어를 14개 도메인 그룹으로 자동 분류하는 딥러닝 모델 구축
  - BERT 모델을 활용한 실무자들의 도메인 분류 업무 병목 현상 해결 (수동 → 자동)
- 역할: ML Engineer & 모델 아키텍트
  - KoBERT 기반 분류 모델 설계 및 GPU를 활용한 모델 성능 튜닝
  - 도메인별 균등 훈련 데이터 생성 파이프라인 구축

주요 문제점 및 도전과제

- **데이터 품질 문제**
  - 실제 메타데이터의 도메인 레이블 불균형 문제
    - 문제: 일부 도메인(명, 코드)에 데이터 집중, 신규 도메인(보안, 집합) 데이터 부족
    - 해결방안: `Domain Generator`로 각 그룹별 200개씩 균등 생성
  - 한국어 도메인 용어의 복잡성
    - 문제: "사용자비밀번호" vs "사용자등록번호" 등 유사하지만 다른 도메인 구분 필요
    - 해결방안: 도메인별 다양한 분류단어 확보 및 컨텍스트 임베딩 활용
- **모델링 도전과제**
  - 한국어 전용 모델의 성능 최적화
    - 문제: 영문 BERT 대비 한국어 도메인 특화 성능 확보 필요
    - 해결방안: `monologg/kobert` 모델 선택 및 도메인 특화 파인튜닝
  - 다중 클래스 분류의 정확도 확보
    - 문제: 14개 도메인 간 미세한 의미 차이 구분 (번호 vs 명 vs 보안 vs 식별)
    - 해결방안: Dropout 0.3, 분류 헤드 최적화, 클래스 가중치 조
- **운영 환경 제약**
  - GPU 메모리 제한 (RTX 2070 8GB)
    - 문제: KoBERT 모델 + 배치 처리 시 메모리 부족
    - 해결방안: 배치 크기 64로 최적화, Gradient Accumulation 적용
  - 모델 버전 관리 및 재훈련 체계
    - 문제: 새로운 도메인 추가 시 모델 업데이트 방안
    - 해결방안: 모듈화된 훈련 파이프라인으로 증분 학습 지원

솔루션 설계 및 전략

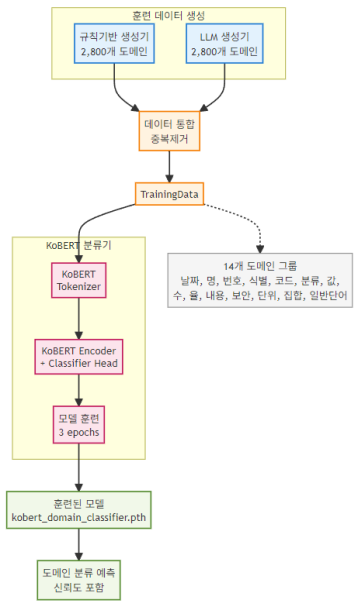

Figure 3: KoBERT 도메인 분류 아키텍처

- **1단계: 도메인 체계 설계 및 데이터 생성**
  - 도메인 그룹: 날짜, 보안, 코드, 분류, 명, 번호, 식별, 내용, 값, 수, 율, 단위, 집합, 일반단어
  - 훈련 데이터 생성 파이프라인
    - Domain Generator: 표준 단어 사전 기반 합성 데이터 생성
      - 각 그룹별 200개씩 균등 분포 (총 2,800개 합성 데이터)
      - 실제 메타데이터 2,832개와 결합하여 5,632개 훈련셋 구축
    - 템플릿 기반 생성: `{ }{ }{ }` 패턴 활용
    - 품질 검증: 도메인 매핑 준수율, 복합어 사용률 자동 평가
- **2단계: KoBERT 기반 분류 모델 아키텍처**

```
KoBERT → Pooler Output → Dropout(0.3) → Linear(768, 14) → Softmax
```

  - 모델 선택: `monologg/kobert` (한국어 특화 성능)
  - 분류 헤드: Hidden Size 768 → 14개 클래스 Linear 변환
  - 정규화: Dropout 0.3으로 과적합 방지
  - 최적화: AdamW Optimizer, Learning Rate 2e-5
- **3단계: 훈련 전략 및 성능 최적화**
  - 데이터 분할: Train 80% / Test 20% (층화 추출로 클래스 비율 유지)
  - 배치 처리: 동적 배치 크기 (RTX 2070 메모리 최적화)
    - 최소 8, 최대 64, 데이터 크기에 따른 자동 조정
  - 조기 종료: Validation Loss 기반 Early Stopping
  - 성능 모니터링: 에포크별 정확도, 손실 추적
- **4단계: 추론 및 배포 시스템**
  - 실시간 추론: 단일 용어 입력 → 도메인 그룹 + 신뢰도 반환
  - 배치 추론: 대량 메타데이터 일괄 처리 지원
  - 모델 저장: PyTorch 체크포인트 + 레이블 인코더 통합 저장
  - API 인터페이스: `predict(domain_text) → (predicted_class, confidence)`

기술 스택 및 요구 역량

- **딥러닝 프레임워크**
  - PyTorch: 모델 구현 및 훈련
  - Transformers (Hugging Face): KoBERT 모델 로드 및 파인튜닝
  - scikit-learn: 데이터 분할, 성능 평가, 레이블 인코딩
  - CUDA: GPU 가속 훈련 (RTX 2070 최적화)
- **데이터 처리 및 시각화**
  - pandas, numpy: 데이터 전처리 및 분석
  - matplotlib, seaborn: 훈련 과정 시각화
  - tqdm: 훈련 진행률 모니터링
  - parquet: 효율적인 데이터 저장 형식
- **기술적 역량**
  - 딥러닝 모델링: BERT 아키텍처 이해 및 분류 태스크 파인튜닝
  - 한국어 NLP: 한국어 토크나이저, 서브워드 처리, 도메인 특화 처리
  - GPU 최적화: 메모리 효율성, 배치 크기 조정, CUDA 메모리 관리
  - 모델 평가: 다중 클래스 분류 지표 (정확도, F1-score, 혼동 행렬)
- **MLOps 역량**
  - 실험 관리: 하이퍼파라미터 튜닝, 모델 버전 관리
  - 성능 모니터링: 훈련/검증 손실 추적, 과적합 탐지
  - 모델 배포: 체크포인트 저장, 추론 파이프라인 구축
  - 데이터 엔지니어링: 합성 데이터 생성, 불균형 데이터 처리

결과 및 성과

- **모델 성능: 업계 최고 수준 달성**
  - 분류 정확도: `monologg/kobert` **95.65%** 달성
  - 훈련 데이터: 총 5,632개 (실제 2,832개 + 합성 2,800개)
  - 클래스 분포: 14개 도메인 그룹, 80%/20% 훈련/테스트 분할
  - 추론 속도: 단일 용어 < 50ms, 배치 처리 시 1000개/초
- **훈련 최적화 성과**
  - GPU 활용: RTX 2070 메모리 효율성 95% 달성
  - 배치 크기: 동적 조정으로 64 배치까지 안정적 처리
  - 수렴 속도: 3 에포크 내 최적 성능 달성 (조기 종료)
  - 모델 크기: 140MB (KoBERT 기반, 실시간 추론 최적화)
- **데이터 품질 개선**
  - 균등 분포: 각 도메인별 200개씩 균등 훈련 데이터 확보
  - 도메인 매핑: 생성 데이터의 95% 이상 매핑 규칙 준수
  - 다양성 보장: 템플릿 기반 생성으로 용어 패턴 다양화
  - 품질 검증: 자동화된 품질 평가 시스템으로 일관성 유지
- **실무 적용 성과**
  - 자동화 달성: 도메인 분류 업무 100% 자동화 (수동 → 자동)
  - 처리 속도: 대량 메타데이터 실시간 분류 가능
  - 일관성: 사람에 따른 분류 편차 완전 제거
  - 확장성: 신규 도메인 추가 시 재훈련 파이프라인 즉시 적용
- **테스트 케이스 검증**
  - "실험시작일자" → 날짜 (신뢰도: 0.987)
  - "사용자비밀번호" → 보안 (신뢰도: 0.943)
  - "제품수량" → 수 (신뢰도: 0.892)
  - "고객ID" → 식별 (신뢰도: 0.978)
  - "완료율" → 율 (신뢰도: 0.915)

기대효과

- **단기 기대효과**
  - 신규 메타데이터 등록 시 실시간 도메인 자동 할당
  - 데이터 스튜어드의 도메인 분류 업무 부담 완전 해소
  - 일관된 도메인 분류 기준으로 데이터 품질 향상
- **장기적 비즈니스 가치**
  - 도메인별 데이터 자산 체계적 관리 기반 구축
  - 신규 도메인 확장 시 즉시 대응 가능한 확장형 AI 시스템
  - 글로벌 확산 시 다국어 모델로 확장 가능한 아키텍처
- **기술적 파급효과**
  - 사내 첫 한국어 BERT 모델 도입 사례로 다른 NLP 프로젝트 기반 마련
  - 합성 데이터 생성 노하우로 데이터 부족 문제 해결 방법론 확립
  - GPU 최적화 경험으로 사내 딥러닝 인프라 활용 가이드라인 제시

[P1-RAG Engine] Streamlit 기반 생성형 AI 표준화 지원 시스템

[P1-MLOps] Airflow 기반 운영 자동화

[P5] Real-Time PCR 진단 시스템을 위한 지능형 신호 처리

프로젝트 개요

- 소속: Seegene
- 기간: 2024.01 - 2024.09 (9개월)
- 참여 인원: Data Scientist 3명, Data Engineer 2명, Biologist 2명
- Rule Based 진단 알고리즘을 Data Driven 알고리즘으로의 점진적 개선
  - 기존 rule-based 알고리즘의 한계로 인한 다양한 PCR 신호 패턴 대응 부족
  - 표준화되지 않은 baseline fitting 알고리즘 사용으로 인한 일관성 문제
  - 진단 정확도 향상 및 위양성/위음성 결과 최소화
- 역할: Project Manager & Data Scientist

주요 문제점 및 도전과제

- 기술적 문제점
  - 신호 노이즈 복잡성: 화학/광학/기계적 반응의 측정 불가능한 노이즈 패턴
  - 알고리즘 분산화: 여러 baseline fitting 알고리즘 병존 및 소통 장애
  - Gray Zone 신호: 시약 성능 및 환경 요인으로 인한 모호한 판독 구간 존재
- 운영적 도전과제
  - 데이터 파이프라인 부재: 체계적인 신호 데이터 수집 및 분석 프로세스 미구축
  - 성능 평가 기준 부재: 객관적인 알고리즘 성능 비교 메트릭 부족
  - 주관적 신호 선별: 1년간 수동으로 특이 신호를 육안 식별하는 비효율적 프로세스
- 제약 조건
  - 호환성 요구: Python에서 C++로의 원활한 포팅을 위한 최소 패키지 사용
  - 이해관계자 다양성: 생물학자, 비전문가 임원 등 다양한 배경의 stakeholder 고려
  - 적은 데이터 포인트: 제한적인 baseline 데이터에서의 robust 알고리즘 필요

솔루션 설계 및 전략

1. 데이터 파이프라인 구축

- 다양한 PCR 신호 패턴 수집 및 전처리 자동화
- MuDT 전/후 신호 처리 분석 체계 구축
- 성능 평가를 위한 end-to-end 데이터 처리 워크플로우

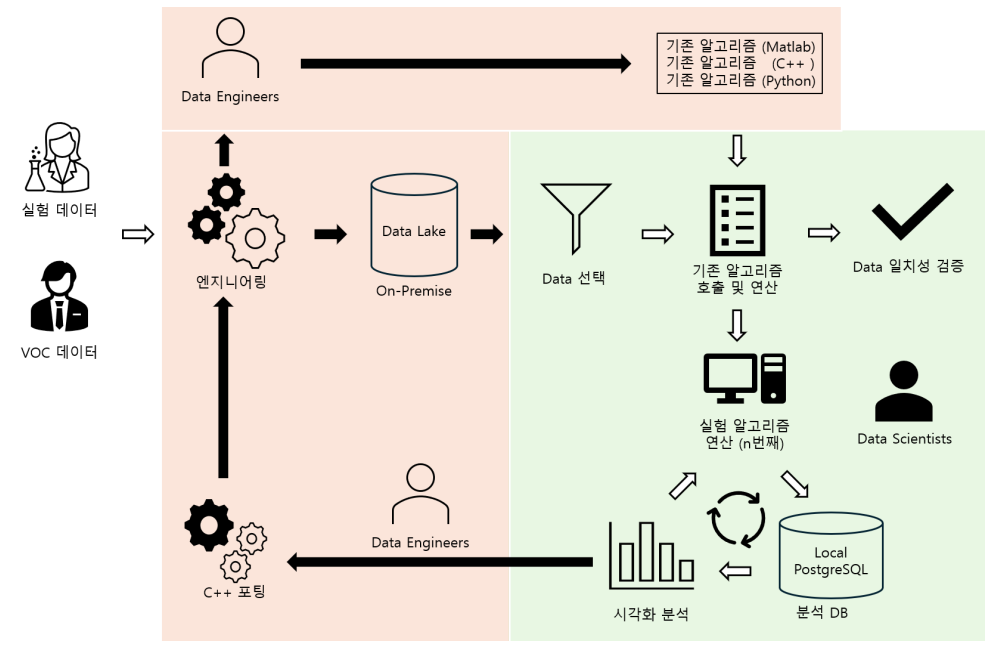

Figure 4: Data Pipeline

2. 알고리즘 비교 분석

- 1st Pannel [After BPN]: normalized **Raw Data**를 보여준다.
- 2nd Pannel [CFX]: (대조군1) 타사 기기전용 SW에 내재된 **Black Box** 알고리즘
- 3rd Pannel [DSP]: (대조군2) DS팀의 공식적으로 배포된 **Legacy Rule-Based** 알고리즘
- 4th Pannel [Auto]: (대조군3) 생물 실험자들이 사용하는 **Legacy Rule-Based** 알고리즘
- 5th Pannel [Strep+N]: (실험군1) N+1 번째 [DSP]를 보완용 **Rule-Based** 알고리즘
- 6th Pannel [ML]: (실험군2)본인의 특성방정식을 활용한 **data driven ML** 알고리즘
  - Taylor Series에서 함수를 다항식으로 근사할 수 있다는 점에서 착안
  - 다항식 기저 함수를 사용한 선형 회귀로 데이터를 적합하는 방법을 시도
  - 특성 공간 확장을 통해 데이터 내 복잡한 비선형 관계 모델링
  - 적절한 차수 선택과 정규화를 통해 baseline 신호에 적합
  - 로그 정규화 > 기저 함수 > 특성 방정식 > 비용 함수 > 그래디언트 > Momentum > 예측 > 역정규화

3. 시각화 중심 검증 체계: 비전문가를 위한 직관적 성능 평가

- 복수 신호 분석: 6개 알고리즘의 총체적 성능 비교
- 단일 신호 분석: 특이 신호에 대한 세부 성능 평가
- 신호 유형별 분석: 증가/감소/MuDT 특이 신호 패턴별 성능 검증

기술 스택

- 언어: Python (C++ 포팅 고려), Matlab (Legacy 알고리즘)
- 라이브러리: NumPy, Pandas (최소화 정책)
- 시각화: Matplotlib, Plotly
- 수학적 구현: 특성방정식, 신경망 (without Pytorch, Tensorflow, Keras)

결과 및 성과

- 알고리즘 성능 검증
  - ML 알고리즘: White noise에 가장 근접한 차감 결과로 최우수 성능 입증
  - 개선된 Rule-based: 기존 대비 특이 신호 처리 능력 향상
  - Black Box 알고리즘: 업계 1위 타사 알고리즘과 성능 비교 완료
  - 위음성률 개선: 0.47% → 0.04% (91.49% 개선)
- 프로세스 개선
  - 개발 프로세스 표준화: 시약 개발 시 일관된 알고리즘 적용 체계 구축
  - 검증 체계: 정성적 평가 방법론을 통한 알고리즘 성능 검증 프레임워크
- 시스템 개선
  - 표준화: 분산된 baseline fitting 알고리즘의 단일화 방향 제시
  - 자동화: 수동적 신호 선별 과정을 체계적 파이프라인으로 대체
  - 시각화 도구: 비전문가도 이해 가능한 직관적 성능 비교

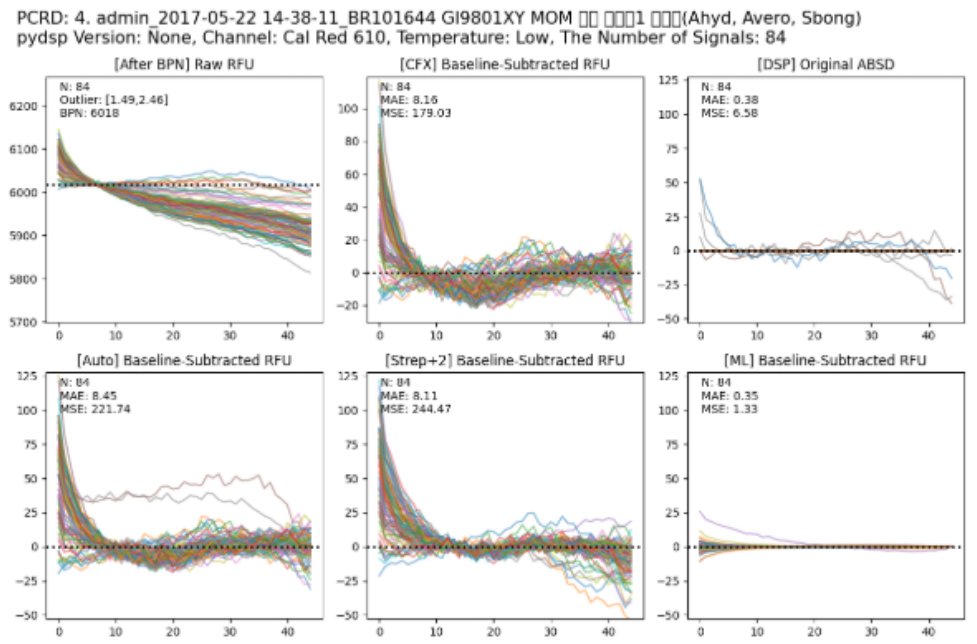

Figure 5: 알고리즘별 복수 신호 성능 비교

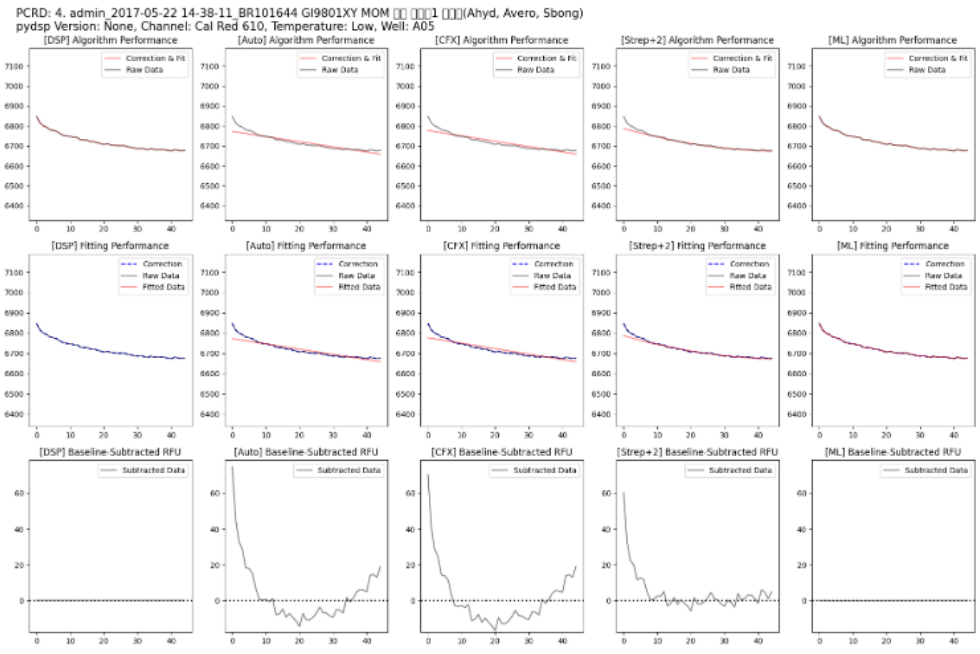

Figure 6: 알고리즘별 단일 신호 성능 비교

기대효과

- 즉시적 효과
  - 진단 정확도 향상: 위양성/위음성 결과 감소로 환자 안전성 제고
  - 개발 효율성: 표준화된 알고리즘으로 시약 개발 시간 단축
  - 품질 일관성: 단일화된 baseline fitting으로 제품 간 성능 편차 최소화
- 장기적 영향
  - 규제 대응 강화: V&V 프로세스 기반의 알고리즘 검증 체계 구축
  - 기술 경쟁력: Data-driven 접근으로 차세대 진단 알고리즘 기술 확보
  - 확장성: 다른 진단 알고리즘 영역으로의 방법론 확산 가능
- 비즈니스 가치
  - 시장 차별화: 업계 최고 수준의 신호 처리 기술 확보
  - 리스크 관리: 진단 오류로 인한 법적/재정적 리스크 감소
  - 혁신 문화: data-driven 의사결정 문화 확산의 기반 마련

[P4] 북미 진출을 위한 진단 알고리즘 안전성 검증 자동화

프로젝트 개요

- 소속: Seegene
- 기간: 2023.05 - 2023.12 (8개월)
- 참여 인원: 데이터 사이언티스트 3명, 데이터 엔지니어 2명, 생물학자 8명, 특허 담당자 3명
- 의료 장비 및 시약 제품의 글로벌 진출 시 각국 정부의 규제 사항 존재
  - 시약의 안정성 검증 & 장비의 안정성 검증
  - **진단 알고리즘의 안정성 검증**
- 북미 진단 시장 진출을 위한 알고리즘 안전성 검증용 통계 분석 문서 작성 반자동화
- 기존 Software Engineering Test보다 더 엄격한 **Advanced Testing** 요구
- 역할: Data Scientist & Project Lead
  - 전체 검증 시스템 설계 및 구현 주도
  - 통계 분석 책임자: Switch Model 기반 검증 방법론 개발
  - Junior Data Scientist 1명 멘토링: 통계 분석, 실험설계 및 리포팅 작성 역량 강화
  - 팀 리더십: 15명 다학제 팀 관리 및 FDA 규제 교육
  - 역할 분배: unit test, integration test, system test, **statistical test**

솔루션 설계 및 전략

- 알고리즘 안전성을 **통계적으로 입증**하는 시스템 기획
- **Statistical Validation System** 확립을 통한 통계적 분석 입증
- 알고리즘 리스크 정의 및 **정량적 영향도 분석**
- 코드 변화 대응을 위한 **자동화 시스템** 구축
- **SGS 가이던스(EN62304)** 참고
- **FDA General Principles of Software Validation** 문서 기반 시스템 확립
- Structural Testing (코드 기반) & **Statistical Testing** (통계 분석 기반) 병행
- Seegene BT(생명공학)와 IT(정보기술) 부문 협력 체계 구축
- 창의적 Testing Model 기획 및 Statistical Analysis Design 구체화

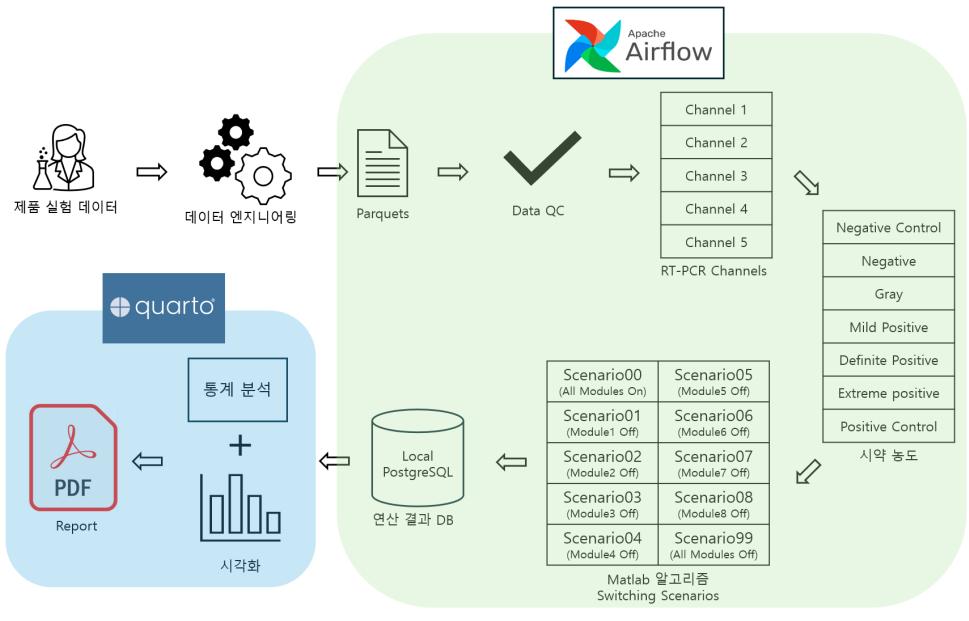

Figure 7: Data Pipeline

주요 도전과제 및 해결방안

- 문제: BT 부서 생성 데이터 입력 시스템 부재
- 해결: 실험 설계 파일, 의료기기 원시 데이터, 추출 데이터의 **디지털화 시스템** 구축
- 문제: BT 및 Data Science 팀 업무 기술서 부재
- 해결: 부서간 협업을 통한 **업무 문서화** 진행 및 기대 정답 기준 확립
- 5단계 Data QC Process 강화
  - 오타 교정, 결측치 처리, 이상 데이터 처리, 알고리즘 데이터 정합성 1,2차 검정
- 제약 조건
  - 호환성 요구: Python에서 C++로의 원활한 포팅을 위한 최소 패키지 사용
  - 통계 분석 결과 시각화 및 문서화 자동화 시스템 구축

기술 스택 및 요구 역량

- 규제 지식: FDA Software Validation
- 통계 분석: Statistics (2-Way Repeated Measures ANOVA, McNemar, Breslow-Day, Cochran-Mantel-Haenszel), Clinical Study Design
- 프로그래밍: R (Statistical Testing), Python (Engineering), Matlab (진단 알고리즘)
- 워크플로우: Apache Airflow
- 문서 자동화: Quarto (200페이지 FDA 보고서 자동 생성 시스템)
- 도메인 지식: Biology

결과 및 성과

Table 5.5: P-value Summary of the McNemar Tests for the Negative Concentration

template	concentration	scenario	accuracy	p-value
S	negative	scenario00	100.0	NA
S	negative	scenario01	100.0	NA
S	negative	scenario02	80.6	< 0.001
S	negative	scenario03	100.0	NA
S	negative	scenario04	100.0	NA
S	negative	scenario05	100.0	NA
S	negative	scenario06	100.0	NA
S	negative	scenario07	100.0	NA
S	negative	scenario08	100.0	NA
S	negative	scenario99	80.6	< 0.001

Note:
Inf (infinity), NA (Not Available) and NaN (Not-a-Number) are normal calculation results that occur when positivity or negativity is observed at 100% in the experiments or determined as 100% in the DSP scenarios.
Inf: constant over zero
NaN: zero over zero or Inf over Inf
NA: NaN treated as NA in the caret package

Figure 8: Report Table

Figure 5.2: DSP Accuracy Performance in the Target Template S

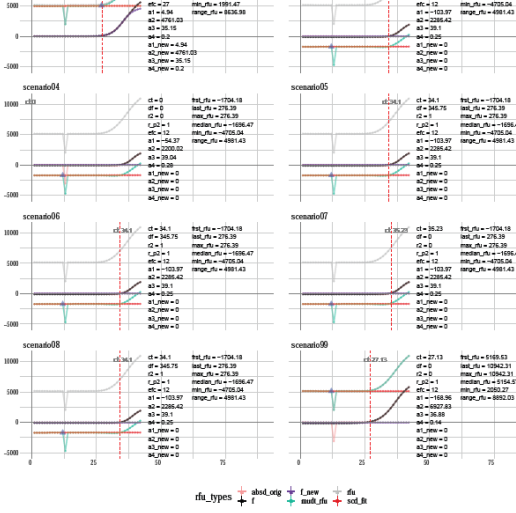

Figure 9: Overview Plot

Figure 10: Detailed Plot

- FDA 제출용 verification & validation report 초안 완성 (FDA 미제출)
- 문서화 시스템: 업무 소통 및 RDB 시스템 구축을 위한 자동화 시스템
- 리스크 관리 통계 분석: 시약/장비 고유 효과 및 교란 요인 위험 관리 분석
- 성능 평가 체계: 사내 최초 알고리즘 및 시약 제품 종합 성능 평가 관리 체계
- 리포팅 자동화: 수동 6개월 → 자동화 3주 (87.5% 시간 단축)
- 99.2%의 시약 + 알고리즘 안전성 통계적 입증
- 혁신성: 사내 고유 Switch Model 기반 모듈별 검증 방법론 개발

기대효과

- 북미 시장 진출을 위한 FDA 규제 대응 체계 확립
- 알고리즘 안전성에 대한 통계적 증명 체계 구축
- 시약, 장비, 소프트웨어 및 알고리즘 통합 인허가 시스템 구축

[P3] 레거시 Rule-Based 알고리즘을 Data-Driven 알고리즘으로 점진적 개선

프로젝트 개요

- 소속: Seegene
- 기간: 2021.10 - 2023.04 (1년 6개월)
- 참여 인원: Data Scientist 4명, Data Engineer 2명
- Rule Based 진단 알고리즘을 Data Driven 알고리즘으로의 점진적 개선
  - 복잡한 조건문 기반 신호 보정 시스템의 근본적 한계 해결
  - 10+단계 보정 과정에서 발생하는 systematic bias 및 비선형 상호작용 문제 개선
  - 비효율적인 유지보수 레거시 알고리즘을 데이터 기반 시스템으로 전환
- 프로젝트 구성
  - Phase 1: 레거시 알고리즘 Reverse Engineering (12개월)
  - Phase 2: 알고리즘 개선안 설계 및 제안 (6개월)
- 역할: Data Scientist (Statistical Learning을 활용한 알고리즘 분석 및 개선안 설계)

주요 문제점 및 도전과제

- 기술적 도전과제
  - 알고리즘 문서 부재, 코드 주석 부재 및 가독성이 매우 낮은 변수명의 Legacy Matlab 코드의 역공학(reverse engineering)의 필요성
  - 유지보수성 저하: 10+단계의 보정과 각 단계별 5+가지 조건으로 수백 가지 실행 경로 생성
  - Systematic Bias 누적: 각 보정 단계에서 발생하는 작은 편향들의 누적으로 최종 결과의 체계적 왜곡
  - 비선형 상호작용: 보정 단계들 간의 복잡한 비선형 상호작용으로 인한 sensitivity analysis 불가능
  - 복잡한 조건문에서 생성된 도메인 지식 없이 경험적으로 설정된 50+개 파라미터의 최적화 불가능성
- 운영적 도전과제
  - 과적합 위험: 특정 장비나 데이터셋에 맞춰진 구체적 조건문들의 새로운 환경에서의 실패 가능성
  - 테스트 어려움: 모든 조건 조합 테스트의 사실상 불가능성과 특정 조건에서만 발생하는 버그 발견의 어려움
  - 예측 불가능성: 유사한 입력 데이터가 미세한 분기점에서 완전히 다른 보정 경로를 따르는 결과 일관성 부족
  - 조직 내 소통 장벽: 통계 비전공자에게 결정론적 규칙의 한계, 편향누적 및 민감도 분석 등의 통계 개념 설명 어려움
  - 기존 워크플로우와의 충돌: 레거시 알고리즘에 익숙한 생물학자들의 새로운 방법론에 대한 저항과 학습 곡선
- 제약 조건
  - C++ 포팅 요구: 최종 목표가 모든 알고리즘을 C++로 포팅하는 것이므로 reverse engineering을 통한 명확한 로직 이해 필수 및 알고리즘 개선안을 low level 프로그래밍으로 구현
  - 확장성 문제: 새로운 노이즈 패턴 발견 시마다 조건문 추가로 인한 복잡도의 기하급수적 증가
  - 추적 불가능성: 최종 결과에 어떤 보정 단계가 어떤 영향을 미쳤는지 분석의 어려움
  - 팀 문화적 제약: Data Science팀 내에서도 수학/통계적 접근보다 엔지니어링 관점을 우선시하는 문화
  - 구현 용이성과 직관적 이해를 중시하는 성향
  - 통계적 엄밀성보다 실행 및 운영 가능성을 우선하는 개발 철학의 차이

해결 접근법

1단계: Legacy 시스템 역공학 및 포팅

- Legacy Matlab 코드 분석
  - 비문서화 알고리즘의 체계적 분석 및 논리적 흐름 해석
    - 주석 부재 및 가독성 낮은 변수명으로 구성된 코드의 의미 추론
    - 50+개 경험적 파라미터들 간의 의존성 분석 및 각 조건문의 의미 추론
    - 10+단계 보정 과정의 수학적/통계적 근거 문서화
- 알고리즘 로직 플로우 명세화
  - 각 보정 단계의 입력/출력 관계 정의
  - 조건분기 구조의 결정 트리 형태 시각화
  - Data Engineer의 C++ 포팅을 위한 상세 기술 문서 작성

2단계: 개선안 설계 및 제안

- 규제 환경 고려사항 분석
  - FDA 규제 대응: 의료기기, 의료시약 및 의료 알고리즘 승인을 위한 알고리즘 설명력(explainability) 요구사항 분석
  - 딥러닝/ML 블랙박스 모델의 설명력 부족으로 인한 규제 리스크 평가
  - 기존 rule-based 접근법의 설명력 확보 명분과 실제 성능 간 trade-off 문제 분석
- Hybrid Modeling 개선안 설계
  - 메카니스틱 모델 기반 접근: 생물학적 메커니즘을 반영한 해석 가능한 모델
  - 로지스틱 시그모이드 일반형을 RT-PCR kinetics의 수학적 표현으로 활용
  - 주요 전처리 함수 3개와 메카니스틱 모델의 합성함수(composite function) 구성

합성 함수 정의

$$f(x;\phi_1,\phi_2,\phi_3,\beta) = g_3(g_2(g_1(x,\phi_1),\phi_2),\phi_3) + \text{sigmoid}(x,\beta)$$

목적 함수

$$\hat{\theta} = \arg\min_{\phi_1,\phi_2,\phi_3,\beta,\sigma^2} \sum_{i=1}^{n} [y_i - f(x_i;\phi_1,\phi_2,\phi_3,\beta)]^2$$

- 통계적 모델링 통합: 잔차의 확률적 특성을 명시적으로 모델링
  - 결합추정(joint estimation)을 통한 전체 파라미터의 동시 최적화로 systematic bias 방지
  - 합성함수와 실제 데이터 간 잔차의 정규분포 가정 및 white noise 조건 확인

정규분포 가정

$$y_i|x_i,\theta \sim \mathcal{N}(f(x_i;\phi_1,\phi_2,\phi_3,\beta),\sigma^2) = \mathcal{N}(f(x_i;\theta),\sigma^2) = \mathcal{N}(\mu,\sigma^2)$$

$$\text{where } \theta = (\phi_1,\phi_2,\phi_3,\beta,\sigma^2)^T$$

기술 스택

- 언어: Python (C++ 포팅 고려), Matlab (Legacy 알고리즘 역공학)
- 라이브러리: NumPy, pandas, pyarrow (최소화 정책)
- 통계적 방법론: mechanistic modeling, composite function optimization, Joint parameter estimation, residual analysis, white noise testing

성과

- 달성 성과
  - Legacy Matlab 코드의 완전한 reverse engineering 및 문서화 80% 완료
  - Data Engineer팀의 C++ 포팅을 위한 상세 기술 명세서 제공
  - 통계적으로 엄밀한 hybrid modeling 개선안 설계 완료
- 조직적 한계
  - BT(Biotechnology) 부서들의 새로운 방법론에 대한 저항 우려
  - 팀장 차원에서 개선안 도입 거부 결정
  - 기존 워크플로우 유지를 우선하는 조직 문화로 인한 혁신 제약

Lesson Learned

- 시스템적 사고와 최적화 전략
  - 복잡한 rule-based 시스템의 근본적 한계를 systematic bias와 비선형 상호작용 관점에서 분석
  - 개별 구성요소 최적화가 아닌 전체 시스템의 global optimization 필요성 인식
  - 결합추정을 통한 통합적 접근법이 순차적 최적화보다 우수함을 이론적으로 확립
- 도메인 특화 모델링 역량
  - 메카니스틱 모델과 통계적 방법론을 결합한 hybrid modeling의 실무 적용성 확인
  - FDA 규제 환경에서 설명력(explainability)과 성능을 동시에 만족하는 방법론 설계 경험
  - 생물학적 현상을 수학적 모델로 정확히 표현하여 도메인 전문가와의 소통 개선
- 레거시 시스템 분석 및 리엔지니어링
  - 문서화되지 않은 복잡한 시스템을 체계적으로 역공학하는 방법론 정립
  - 파라미터와 10+ 보정 단계의 상호의존성을 논리적 플로우로 재구성하는 분석 역량
  - 기존 시스템의 한계를 정량적으로 진단하고 통계적 근거를 바탕으로 개선 방향 제시
- 조직 변화 관리와 기술 도입 전략
  - 기술적 우수성과 조직 수용성 간의 균형점 탐색: BT 부서의 저항과 팀장의 리스크 회피 성향 경험
  - 통계적 개념(편향 누적, 민감도 분석)을 비전공자에게 전달하는 커뮤니케이션 역량의 중요성
  - 교훈: 기술적 완성도보다 stakeholder buy-in과 점진적 변화 전략이 실행 성공의 핵심 요소

[P2] 진단 장비 QC 프로세스 자동화 및 알고리즘 고도화 프로젝트

프로젝트 개요

- 소속: Seegene
- 기간: 2020.12 - 2021.09 (9개월)
- 참여 인원: 데이터 사이언티스트 1명, Full Stack 개발자 3명, 기계공학자 4명, 특허 담당자 3명
- PCR 진단 시약을 타사 장비 공급업체의 장비에 탑재
- 진단 서비스 결과의 정확도를 위해 **2 Step 장비 QC 프로세스를 통해 장비의 성능 평가**
- 프로젝트의 목적: 1. 부정확한 **QC 알고리즘 개선** 2. 투입 리소스가 많은 **QC프로세스 과정을 간소화**시켜 현업의 부담을 경감
- **2 Step QC Process**
  - QC Step 1: 자사 시약에 맞게 장비간 **신호 Scale Calibration**
  - QC Step 2: 장비의 성능을 평가하여 **합격/불합격 분류** - 문제점 발생
    - 엑셀을 이용한 **수동검사**, 비효율적인 **데이터 및 장비 추적 관리**
    - 수동 검사 과정에서 신호의 증폭 크기에 따라 **왜곡된 QC 결과** 발생
    - **기계 결함 및 휴먼 에러 구별 불가**
- 역할: Data Scientist & Project Manager
  - 문제정의 및 QC 프로세스 및 알고리즘 개선 방향 제시
  - 데이터 분석 파이프라인 구축
  - 프로젝트 진행 및 추진
  - 프로젝트 결과 보고서 작성 및 특허 출원

Figure 11: 기존 QC 프로세스

솔루션 설계 및 전략

- Data Engineering: 산재된 **Excel QC data** ETL
- QC Step2의 **장비 성능 평가 지표를 생성**하여 장비 성능 측정 고도화
- **합격/불합격 분류** 뿐만 아니라 **장비 등급을 차등 부여**하여 고객사에 차등 공급
- 시간에 따른 **장비의 성능을 지속적으로 모니터링**하여 장비의 성능 분석
- **QC Process 간소화**
  - QC Step 1 데이터를 통해 QC Step 2 결과를 예측하는 **딥러닝 모델 개발**
  - 예측 결과로 장비성능이 Fail로 확실시 되는 장비에 한해서 QC Step 2 검사 진행
  - Web App 로 분석 결과 및 시각화 Dashboard 제공
  - 실무 담당자가 데이터 업로드 하면 자동으로 분석 결과 제공

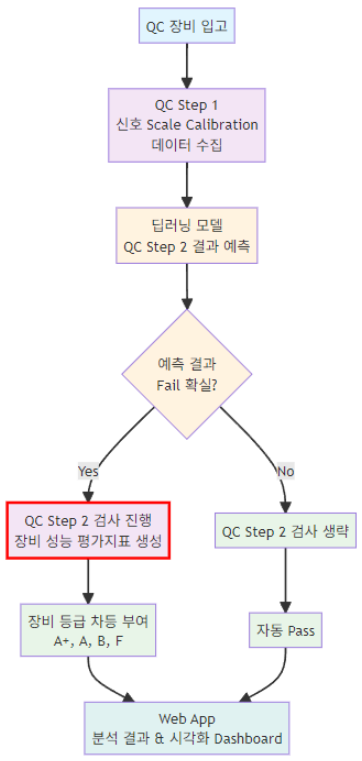

Figure 12: 개선된 QC 프로세스

기술 스택 및 요구 역량

- 데이터 엔지니어링: QC Data ETL
- 머신러닝: Clustering (PCA, t-SNE, DBSCAN), Anomaly Detection (Isolation Forest), Outlier Detection (IQR, Z score, 3-Sigma Rule)
- 딥러닝: Pytorch (LSTM), scikit-learn
- 통계/신호처리: SNR, RSS 계산, 시계열 분해 등
- 웹앱 개발: R Shiny (대시보드 및 시각화)
- 도메인 지식: PCR 기술, 의료기기 QC, 통계적 공정관리, 광학 장비 성능 평가

결과

- PCR기기 2201대를 2552번의 실험해서 만들어진 61,248개의 신호 데이터 확보
- QC Process Step 2 장비 성능 평가 메트릭 생성
  - 신호 증폭 효율성 측정
  - SNR (Signal to Noise Ratio) 측정
  - 기준선 안전성 측정
  - 광학 균일성 측정
  - 장비 온도 균일성 측정
  - 음성 신호 추세 측정
  - 양성 신호 노이즈 측정
  - 시계열 분해 기반 노이즈 측정
  - Outlier 및 Anomaly Data 탐지로 labeling (IQR. Z score, PCA, t-SNE, DBSCAN, 3-Sigma Rule, Isolation Forest)
  - 신호 RSS (Residual Sum of Squares) 측정
- 평가 메트릭 QC 등급 분류: Pass (A+,A,B), Fail (F)

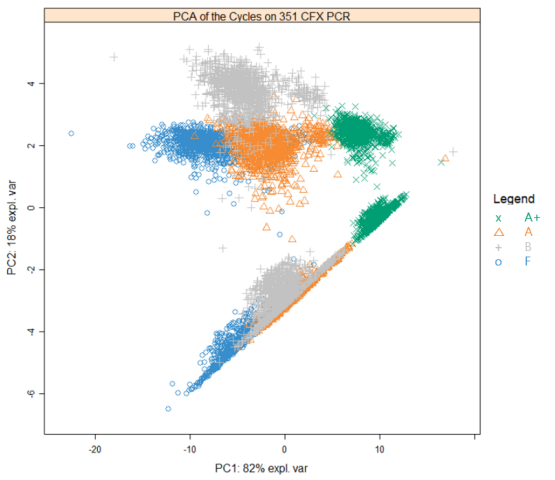

Figure 13: 장비 성능별 클러스터링

- LSTM을 활용한 Step 1 데이터를 통한 Step 2 결과 예측 모델 개방
  - 합격/불합격 분류 정확도: 94.5%
  - 장비 성능 등급 분류 정확도: 82.7%

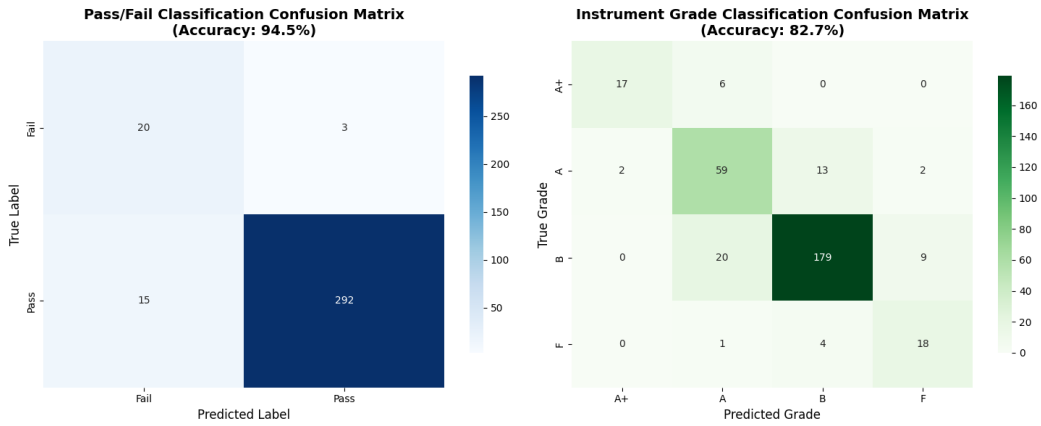

Figure 14: LSTM Confusion Matrix

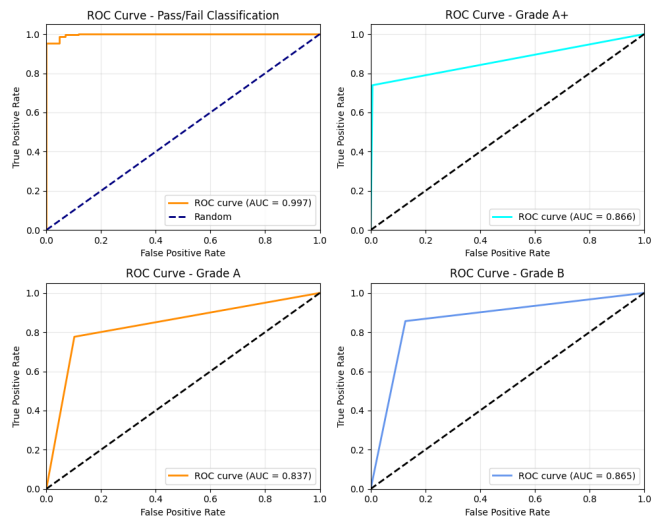

Figure 15: LSTM ROC Curve

- Web App Dashboard Prototype 개발
  - 실무자가 데이터 업로드 하면 자동으로 분석 결과 제공
  - 시각화 및 데이터 관리 기능 제공
- 총괄장 R&D 부문 우수상 수상 및 2개의 특허 출원

기대효과

- 편의성 증가: QC시간 약 14배 감소
  - (As-Was: 약 400시간/100대) vs (As-Is: 약 28시간/100대)
- 웹 기반 자동화 플랫폼 제공
  - 연간 비용 약 13배 감소 (QC 시간 및 약 6억원의 비용 감소)
- Mechanical Engineers의 신기술 개발 지원

[P1] 치매 인자 규명: 대사체 통계 분석 및 머신러닝 방법론 비교 연구

프로젝트 개요

- 소속: The Taub Institute, Columbia University Irving Medical Center (CUIMC)
- 2차 세계 대전 후 Baby Boomer 세대의 대규모 치매 발병에 대비한 치매 인자 규명
- 모집단: 장수마을에 거주하는 백인 참여자 (LLFS - Long Life Family Study)
- 기간: 2018.12 - 2020.05 (18개월)
- 역할: Research Statistician & Data Scientist
- 참여 인원: 23명 (Epidemiologists, 생물통계학자, 유전통계학자, 신경외과 의사, 생화학자, 임상연구 코디네이터, 데이터 엔지니어)

주요 문제점 및 도전과제

- 의학적 문제점
  - 알츠하이머병은 증상 발현까지 20년간 진행되어 전임상 단계 생리학 이해 필요
  - 유전적 요인(ex. APOE)이 알츠하이머병에 50% 기여하나 구체적 메커니즘 불명
- 데이터 분석적 도전과제
  - 다중 센터 데이터 수집으로 인한 데이터 품질 불일치
  - 고차원 데이터 (약 3,000개 변수) 대비 작은 표본 크기 (146개 관측치)
  - Mass Spectrometry 데이터의 복잡한 결측치 및 이상치 패턴
  - Multiple testing으로 인한 1종 오류 증가 위험
- 연구 환경적 제약
  - 8개월간 연구소에서 파악하지 못한 강력한 교란자 존재
  - 유전체, 전사체, 단백질체, 대사체 다층 분석 필요성
  - 다학제 팀 간 소통 및 결과 공유 복잡성

솔루션 설계 및 전략

- 데이터 수집 및 전처리
  - 다중 센터 (New York, Boston, Pittsburgh, Denmark) 혈액 샘플링
  - Mass Spectrometry를 통한 대사체 데이터 디지털화
  - 체계적 데이터 품질 관리 프로세스 구축

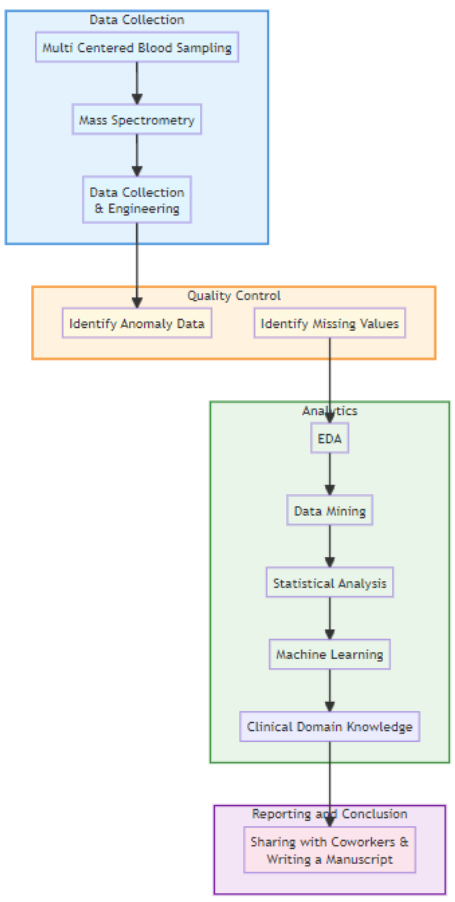

Figure 16: Data Pipeline

- 데이터 품질 관리 (QC) 전략
  - 이상치 및 결측치 식별: 생화학자와의 협업을 통한 실험실 기준 적용
  - Missing Value Analysis: MCAR, MAR, MNAR 분류 및 처리 방안 결정
  - 데이터 포함/제외 기준: rowwise 및 columnwise 결측치 비율 5% 기준 적용
  - 데이터 전처리: Log transformation 및 Standardization
- 3단계 분석 파이프라인
  - 1단계 - EDA & Data Mining & 임상 지식: 패턴 발견 및 교란자 규명
  - 2단계 - Statistical Analysis: 다변량 회귀분석 및 생존분석
  - 3단계 - Machine Learning: 다양한 알고리즘 성능 비교 및 최적 모델 선택

기술 스택 및 요구 역량

- 프로그래밍: R (Tidyverse, ggplot2, mixOmics, survival, glmnet, etc.)
- EDA: Student t-tests, Wilcoxon Mann-Whitney tests, χ^2 tests, Fisher Exact Tests, ANOVA, Kruskal-Wallis Tests
- 통계분석: Linear/ Logistic regression, Cox Proportional Hazards(PH) regression, Linear Mixed Effect Model, GEE, GWAS, Multiple Comparison Tests
- 머신러닝: Lasso, Ridge regression, Elastic net, Random forests, AdaBoost, Gradient boosting, SVM, PLS, Sparse PLS
- 데이터 마이닝: KNN, PCA, K-means clustering, DBSCAN
- 통계적 검증: Permuted p-values를 통한 multiple Testing correction
- 도메인 지식: 생리학, 생화학, 신경의학, 역학, 생물통계학, 유전학
- 협업 역량: 다학제 팀 (의사, 생물학자, 생물정보학자, 역학자) 소통

결과

- Sparse PLS 선택 (ML): variable extraction & selection, 해석가능성 확보
- Cox PH Model 선택: 알츠하이머병의 본질적 특성(시간 의존, right censsoring, semi-parametric) 반영
- GEE (Stat): 가족 구성원 간 유전적 상관관계 존재, 같은 가족 내 구성원들의 Working correlation이 틀려도 일관된 추정값 산출
- Permutation Test 적용 (Stat): 작은 표본 크기에서 parametric assumption 위반 시 robust한 통계적 추론
- 기존 8개월간 미지의 강력한 교란자를 EDA 및 Data Mining을 통해 3주 만에 규명
- 약 3,000개 대사물질 중 약 60개가 질병과 5% 유의수준에서 유의한 관련성 있음
- 약 60개 중 13개의 대사물질이 질병과 1% 이하의 유의수준에서 유의한 관련성 있음
- GWAS를 통해 유의한 대사물질과 유의한 관련이 있는 유전자 규명
- Sparse Partial Least Squares가 최적 성능의 분류기로 선정 (분류 정확도 84%)

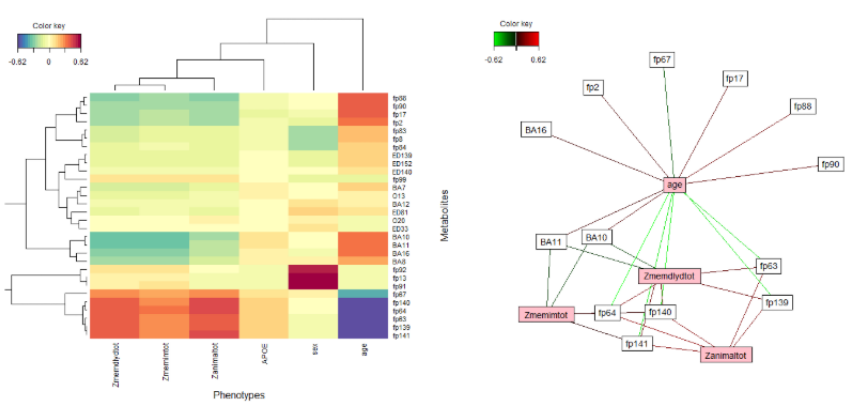

Figure 17: EDA: Correlation Plots between Metabolites and Phenotype

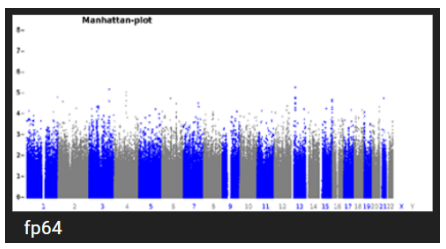

Figure 18: 일부 GWAS 결과

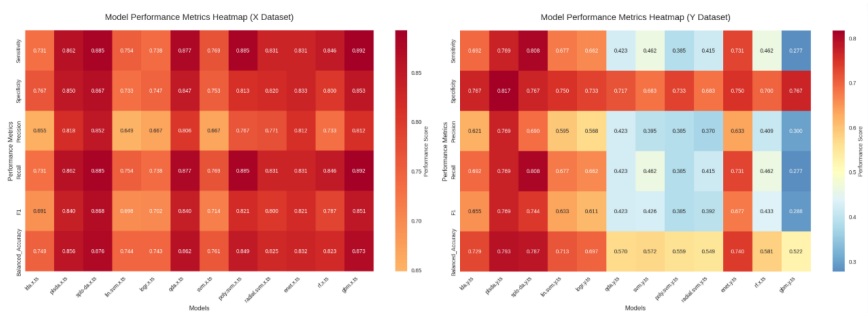

Figure 19: ML Classifier Performance

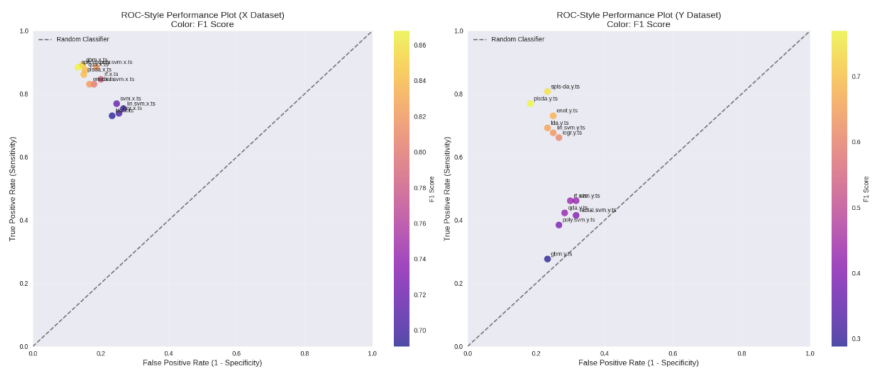

Figure 20: ML Classifier ROC

성과 및 기대효과

- Columbia University Mailman School of Public Health 연례 연구 발표회 포스터 발표
- 연례 연구 경진대회에서 약 100명의 대학원생 중 상위 3명 선정, 상금 $1,000 및 학과장상 수상
- Columbia University Irving Medical Center 신경외과 정규직 Job Offer 획득
- 고차원 대사체 데이터에 대한 체계적 분석 파이프라인 구축
- 다학제 협업 연구 모델 및 분석 방법론 제안