

1. Team Info

1.1 과제명

<머신러닝 기반 베이커리 수요예측 시스템: Randy’s Donuts과의 공동 프로젝트>

본 산학 협력 프로젝트는 Randy’s Donuts 국내 가맹점을 직접 대상으로 하여, 점포 운영 과정에서 반복적으로 발생하는 품질 및 폐기 문제를 동시에 완화하는 것을 목표로 한다. Randy’s Donuts는 본 프로젝트의 **핵심** 협력 파트너이자 실증 대상 기업으로, ~~초기 적용을~~ **본 실험을** 통해 검증된 성과는 향후 베이커리 산업 전반의 수요예측 혁신 사례로 확장 가능하다.

1.2 팀 정보

본 프로젝트는 **13팀, e시크**에서 수행하였다.

1.3 팀 구성원

이름(학번)	역할	역할 설명
이지원 (2176297)	팀 리더(PM), 프론트엔드 엔지니어	프로젝트 매니저(PM)로서 프로젝트 기획 및 일정 관리, 협력사 커뮤니케이션 총괄, 웹 시각화 및 문서화 작업 주도
김도희 (2170009)	AI & 백엔드 엔지니어	데이터 수집 및 전처리, 외부 API(Open-Meteo 등) 연동, 데이터베이스 파이프라인 개발
변아연 (2271032)	AI & 백엔드 엔지니어	머신러닝 모델 학습 및 하이퍼파라미터 튜닝, 성능 평가, 분석 리포트 작성

2. Project Summary

2.1 문제 정의

[Target Customer] 본 프로젝트의 주요 타겟 고객은 베이커리 점포 운영자이며, ~~현재는~~ 협업사인 Randy’s Donuts 국내 가맹점을 대상으로 실증 적용을 진행하고 있다.

[Pain Point] 현재 다수의 베이커리 점포에서는 발주량을 과학적 근거 없이 점주의 경험과 직관(즉 '감')에 의존해 결정한다. 이러한 방식은 구조적으로 다음과 같은 문제를 초래한다.

- 1. 과잉 발주: 판매되지 못한 제품이 폐기되어 불필요한 비용 발생

2. 과소 발주: 예상치 못한 품질로 인한 매출 기회 상실

[Pain Point에 대한 근거] 2025년 7월, 수도권 소재 베이커리 점포 점주들을 대상으로 대면 인터뷰를 진행하였다.

점포	응답 내용
파리바게뜨 의왕포일2지구점, 파리바게뜨 인덕원역점 점주	"발주를 매일 감으로 정하다 보니, 신경을 써도 매일 폐기로 인한 손실이 커서 걱정입니다."
던킨도너츠 판교테크노밸리점 점주	"AI로 수요예측이 잘된다면 쓰고 싶습니다."

그 결과, 베이커리 점포 운영자들이 실제로 과잉 발주와 과소 발주 문제를 경험하고 있으며, 동시에 AI 기반 수요예측 서비스에 대한 활용 의지가 존재함을 확인하고, 본 프로젝트의 필요성과 수요를 검증하였다. ~~또한, 이러한 조사 결과는 베이커리 점포 운영자들이 발주 의사결정 과정에서 경험과 직관에 의존함으로써 발생하는 비효율성을 잘 보여준다. 즉, 품질은 매출 손실을, 폐기는 비용 부담을 초래하며, 이는 실제 점포 운영에 부담으로 작용한다.~~

특히 해당 프로젝트에 관해 링크드인에 게시한 글을 보고 **아** 문제의 심각성에 공감한 (주)라포르엘 Randy's Donuts 측에서 연락을 주었으며, 이를 계기로 협업을 진행하게 되었다. 이는 곧 업계 현장에서 해당 pain point가 실질적으로 존재하며, 본 프로젝트의 접근 방식이 타겟 고객에게도 필요한 해결책임을 방증한다.

따라서 본 프로젝트는 머신러닝 기반 AI 수요예측 시스템을 도입하여 발주 의사결정 **과정**을 데이터 기반으로 전환하고, 품질과 폐기를 동시에 줄여 점포 운영 효율성을 향상시키는 것을 목표로 한다.

2.2 기존 연구와의 비교

서비스 이름	서비스 내용	장점	단점
뚜레쥬르 (CJ푸드빌)	전국 점포별 판매 데이터를 기반으로 다음날 빵 수요를 예측하는 AI 모델 도입. 현재 일부 점포에서 시험 운영 중.	1. 도입 점포에서 매출이 평균 5% 상승 2. 재고과잉 및 품질 현상 감소	1. 소수 점포 대상 시범 적용으로, 범용성 검증 부족 2. 구체적 성능 지표 (R^2 등) 미공개로 기술적 검증 한계
SPC·롯데제과 빅데이터 기반 수요예측	판매 데이터를 활용해 발주 및 유통 효율화를 목표로 빅데이터 분석 시스템 구축.	1. 대규모 데이터를 활용한 통합 관리 가능 2. 물류·유통 단계까지	1. 개별 점포 단위 최적화 부족 (점포 특성·외부변수 미흡)

도입 시도		연결되어 전사 최적화 효과 기대	2. 현장 점주보다는 본사 차원의 관리 지향
글로벌 SaaS 기반 수요예측 솔루션 (예: Blue Yonder)	머신러닝 기반 SKU 단위 수요 예측을 제공하는 글로벌 SaaS 플랫폼	1. 글로벌 산업 전반에서 검증된 알고리즘 보유 2. 자동화·최적화 기능 풍부	1. 비용이 매우 높아 국내 프랜차이즈 점포 단위 적용 어려움. 2. 국내 점포 데이터와 직접적인 적합성이 낮음 (POS, 지역 특수성 부족)

반면, 본 프로젝트는 기존 서비스와 다음과 같은 차별성을 가진다.

첫째, 실제 협업 점포 데이터를 기반으로 검증을 수행한다는 점이다. Randy's Donuts 국내 4개 점포(제주 애월점, 연남점, 광안리점, 수원 타임빌라스점)의 POS 데이터를 활용하여 **각 점포 단위**에 최적화된 모델을 구축하였으며, 협업 점포에서 평균 R^2 0.7~0.8 수준의 예측 성능을 달성하였다.

둘째, 단순 판매 데이터에 그치지 않고, 날씨(~~Open-Meteo API~~), 공휴일, 유동인구, 프로모션과 같은 **외부 변수를 결합함으로써 점포별 특성을 반영**한 정교한 수요예측을 수행한다.

셋째, 현장 적용 가능성 측면에서도 차별성을 확보하였다. 예측 결과는 매주 CSV 리포트 형태로 제공될 뿐 아니라, JAVA 기반으로 개발 중인 웹 대시보드를 통해 점주와 운영자가 직접 조회·활용할 수 있도록 구현되고 있다. 이는 본사 중심의 일괄 관리가 아닌, **프랜차이즈 개별 점포 단위의 의사결정에 즉시 반영 가능한 구조**라는 점에서 기존 서비스와 뚜렷이 구분된다.

2.3 제안 내용

본 프로젝트는 베이커리 점포 운영자를 주요 타겟으로 하며, 점포별 POS 데이터와 외부 변수를 결합한 **일자별·상품별 판매량 예측 시스템**을 제안한다. 머신러닝 기법을 채택한 이유는, 전통적인 통계 모델에 비해 다양한 외부 ~~요인을 변수~~를 반영할 수 있으며, 동시에 대규모 데이터와 고사양 컴퓨팅 자원을 요구하는 딥러닝 모델보다 운영 환경에서 효율적으로 동작하기 때문이다. ~~이를 통해 베이커리 점포 운영자가 발주 의사결정을 데이터 기반으로 내릴 수 있도록 지원하고, 품질 및 폐기 문제를 최소화하는 것을 목표로 한다.~~

구체적인 제안 내용은 다음과 같다.

1. 다변량 데이터 활용

- 점포별 POS 데이터(일자별·상품별 판매 이력)를 기본으로 활용한다.
- 추가적으로 기온·강수량(Open-Meteo API), 요일·공휴일 여부, 유동인구, 점포별 프로모션 이벤트 등 외부 변수를 통합하여 모델 입력값으로 사용한다.

2. 머신러닝 모델 비교 및 최적화

- A. Google Colab GPU 환경에서 CatBoost, HistGradientBoostingRegressor(HGBR) 등 복수의 머신러닝 모델을 적용·비교한다.
 - B. 점포별 학습 데이터 특성을 고려하여, 가장 높은 성능을 보이는 최적 모델을 선정한다.
3. **예측 결과 제공 방식**
- A. **[초기 단계]** 예측 결과를 **CSV 리포트 형태로 제공한다**. 점포별·일자별·상품별 예측치와 권장 발주량을 포함한 리포트를 주 단위로 자동 생성하여 협업사에 전달하며, 발주 참고자료로 활용된다.
 - B. **[중간 단계]** 프로토타입 시연을 위해 **Streamlit 기반 대시보드**를 도입한다. 점포 운영자는 웹 화면에서 예측값과 실제 판매량을 그래프 및 표 형태로 비교할 수 있으며, CSV 파일 다운로드 버튼을 통해 직접 데이터를 확보할 수 있다. 이는 중간 발표에서 **활용될** 데모용 인터페이스다.
 - C. **[확장 단계]** 협업사의 사내 시스템과 연계 가능한 웹 대시보드(JAVA 기반 예측 모듈 + DB 연동)로 발전시킨다. 이 단계에서는 점주가 실시간으로 예측 결과를 조회하고, 점포별 요약·상품별 상세 분석·프로모션 관리 등 다양한 기능을 통해 예측치를 즉각적으로 발주 의사결정에 반영할 수 있다.

2.4 기대 효과 및 의의

본 시스템의 도입을 통해 베이커리 점포 운영자는 다음과 같은 효과를 기대할 수 있다.

1. 폐기 및 품질 문제 완화를 통한 비용 절감 및 매출 안정화

- A. 데이터 기반의 일자별·상품별 판매량 예측을 통해 과잉 발주와 과소 발주를 동시에 줄인다.
- B. 실제 **협업·지점의 협업사의 점포별** 운영 결과, 예측 정확도 ~~$R^2 \geq 0.7 \sim 0.8$ 수준을 달성하여~~ 를 기존보다 향상시켜 폐기·품질 발생 빈도를 유의미하게 감소시킬 수 있음을 확인하였다.
- C. 불필요한 재고 폐기 비용을 줄이고, 품질로 인한 매출 손실을 최소화한다. 이를 통해 점포의 운영 비용을 절감하고, 안정적인 매출 흐름을 유지할 수 있다.

~~2. 비용 절감 및 매출 안정화~~

- ~~A. 불필요한 재고 폐기 비용을 줄이고, 품질로 인한 매출 손실을 최소화한다. 이를 통해 점포의 운영 비용을 절감하고, 안정적인 매출 흐름을 유지할 수 있다.~~

3. 발주 의사결정의 객관성 확보

- A. 기존에는 점주의 경험과 직관에 의존하던 발주 방식을, 예측 모델의 수치 결과를 기반으로 합리적으로 개선할 수 있다. 이는 점포 운영자의 의사결정 부담을 줄이고, 운영 전반의 효율성을 높인다.

이와 같이 본 프로젝트는 **실제 점포 운영의 효율성과 경제적 효과를 동시에 개선할**을 개선함으로써 경제적 이익을 가져올 수 있는 실용적 의의를 갖는다. 나아가, 본사 중심의 관리가 아닌 개별 점포 단위의 맞춤형 수요예측을 가능하게 함으로써 프랜차이즈 운영 모델의 혁신적 사례로 확

장될 수 있다.

2.5 주요 기능 리스트

본 프로젝트는 기능을 **데이터 전처리 자동화, 수요 예측 엔진, 결과 제공, 그리고 웹 대시보드** 제공의 네 가지 축으로 구체화되며, 세부 내용과 구현 현황은 다음과 같다.

기능	기능 설명	현재 구현 여부
1. 데이터 전처리 자동화	1. POS 데이터와 외부 변수(날씨, 공휴일 등)를 결합하여 자동으로 전처리하는 기능 2. 데이터 수집 → 결측치 처리 → 외부 변수 조인까지 전처리 과정 구현 완료 3. 스케줄러 기반 자동화 파이프라인 구축은 진행 중	전처리 구현 완료, 자동화 구현 중
2. 수요 예측 엔진	1. 머신러닝(ML) 모델을 활용하여 일자별, 상품별 판매량을 예측하는 기능 2. CatBoost 모델을 기본으로 적용하고, 일부 점포는 HGBR를 병행하여 최적모델을 구축. 3. SKU 단위 판매량 예측을 지원	모델 구현 완료, 매주 성능 검증 및 재학습 루프 운영 중
3. 결과 제공	1. 점포별, 상품별 예측치와 권장 발주량을 포함한 CSV 리포트를 자동 생성 및 주 1회 협업사에 배포 2. 리포트는 발주 참고용 가이드로 활용 가능	구현 완료 및 운영 중
4. 웹 대시보드	1. 중간 단계: Streamlit 기반 프로토타입 대시보드 운영 <ul style="list-style-type: none">CSV 다운로드 버튼예측값 vs 실제값 비교 그래프상품별 트렌드 조회 2. 확장 단계: JAVA 기반 예측 모듈과 DB 연동 <ul style="list-style-type: none">점포별 요약, 상품별 상세 분석, 이벤트 관리 제공	Streamlit 프로토타입 운영 중, JAVA 기반 대시보드 개발 진행 중

추가적으로, 각 기능의 상세 구현은 다음과 같다.

1. 데이터 전처리 자동화

점포별 POS 원천 데이터와 Open-Meteo API(기온·강수량), 공휴일 DB를 결합하였다. ~~결측치 보정~~ 이상치 제거·범주형 인코딩을 수행하고, POS-외부 변수 병합(join)을 통해 Feature Set을 생성한다. 스케줄러 기반으로 주 1회 자동 수행하는 구조는 현재 ~~구축~~ 개발 중에 있다.

2. 수요 예측 엔진

CatBoost를 기본 모델로 활용하되, ~~데이터 불균형이나 소규모 점포에는~~ 특수 점포(제주애

월점)에는 HGBR을 병행하였다. Hyperparameter Tuning(Grid/Random Search) 및 Feature Importance 분석을 통해 변수 기여도를 평가한다. 모델 성능은 R^2 (목표 ≥ 0.8), MAE 지표로 관리하며, 학습된 모델(.cbm)을 저장해 운영 환경에서 재활용한다.

3. 결과 제공

예측치와 권장 발주량이 포함된 CSV 리포트를 매주 자동 생성한다. 파일명 규칙(날짜 기반), 컬럼 구조, 버전 관리 체계를 적용하여 협업사에 안정적으로 제공하고 있으며, 현재 매주 목요일 정기 배포 체계를 운영 중이다.

4. 웹 대시보드

단계적 발전 전략을 취하고 있다.

- A. **중간 단계:** Streamlit 기반 프로토타입을 통해 예측 결과 시각화, CSV 다운로드, 예측-실제값 비교 그래프, 상품별 트렌드 조회 기능을 제공하여 시연·검증에 활용한다.
- B. **확장 단계:** 협업사 운영 서버와 연동되는 JAVA 기반 대시보드를 구축해 점주가 실시간으로 예측 결과를 조회하고, 이벤트(프로모션 등)를 직접 반영할 수 있도록 한다.

2.6 협업 현황

본 프로젝트는 실제 산업 현장과 협업하여 실증을 진행한다는 점에서 의의가 크다. 현재 (주)라포르엘과 협력하여 Randy's Donuts 국내 4개 점포(제주 애월점, 연남점, 수원 타임빌라스점, 부산 광안리점)을 대상으로 수요 예측 시스템을 운영 중이다.

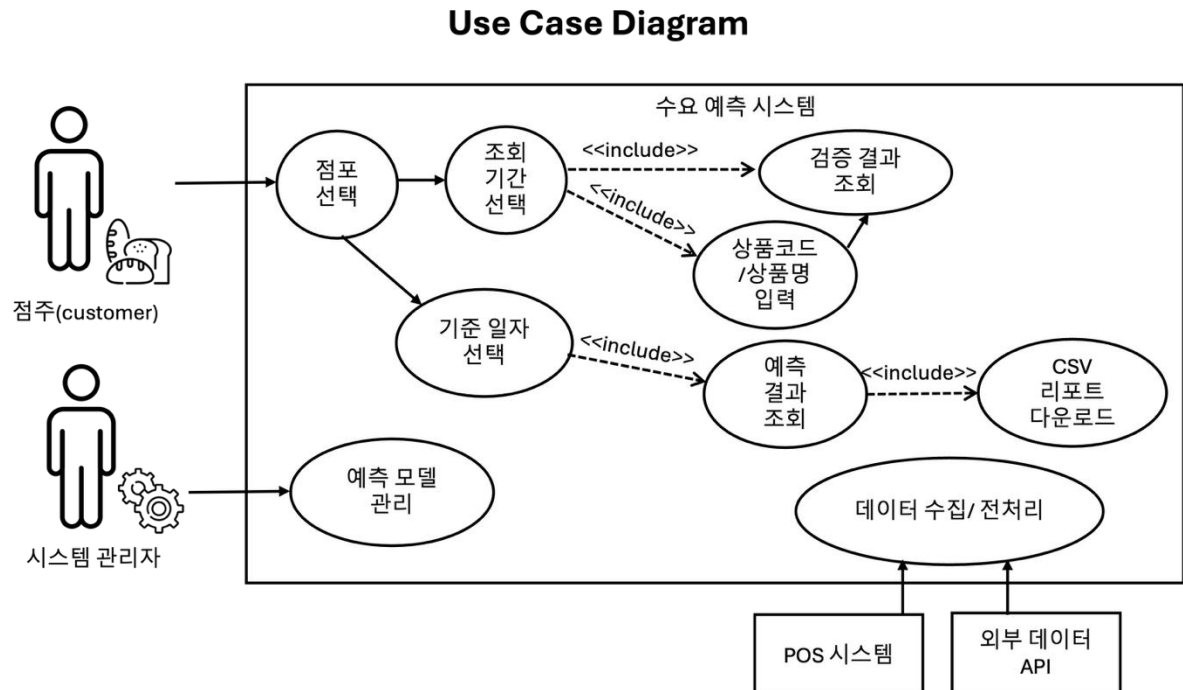
각 점포의 POS 데이터를 기반으로 주 단위 모델 학습 및 검증 루프를 수행하고 있으며, 현재까지 네 점포 모두에서 R^2 0.70~0.80 수준의 예측 정확도를 달성하였다. 점포별 판매 패턴의 이질성을 반영하기 위해 CatBoost를 기본 모델로 적용하되, 일부 특정 점포에는 HistGradientBoostingRegressor(HGBR)를 병행하여 성능을 보완하였다.

또한 협업사 운영 환경에 적합한 엔드투엔드 파이프라인을 구현하였다. 자바 기반 스케줄러가 매주 예측 엔진을 자동 실행하고, 파이썬 학습 코드와 연동하여 모델을 갱신하며, 예측 결과는 중앙 DB에 저장된다. 저장된 결과는 웹 UI에서 실시간 조회가 가능하도록 설계되어, 연구 수준을 넘어 실제 점포 운영 의사결정에 직접 활용 가능한 시스템으로 발전하였다.

3. Project Design

3.1 요구사항 정의

본 프로젝트는 성공적인 수요 예측 시스템 구현을 위해 시스템의 핵심 기능을 네 가지 주요 범주로 나누어 요구사항을 정의하였다. 정의된 요구사항은 아래의 [그림 3.1] use case diagram을 통해 시스템의 사용자 및 외부 요소와의 상호작용 형태로 구체화되었다.



[그림 3.1. 유즈 케이스 다이어그램]

1. 데이터 처리 요구사항

시스템은 예측 모델 학습 및 운영을 위한 신뢰성 높은 데이터 파이프라인을 구축해야 한다. 핵심 데이터 소스인 POS(판매 시점 정보) 데이터뿐만 아니라 예측 성능 향상을 위해 외부 변수를 반드시 연동해야 한다. 이를 위해 Open-Meteo API를 활용하여 지역별 날씨 데이터를 안정적으로 수집~~해야~~ 하며, 모든 수집 데이터는 예측 모델 입력 형식에 맞게 전처리 및 통합 과정을 ~~거쳐야 한다~~ ~~다~~ ~~거친다~~.

2. 모델링 요구사항

시스템은 높은 예측 정확도를 달성하기 위해 머신러닝 기반의 수요 예측 엔진을 구축해야 한다. 기존 평균 예측 방식 대비 예측 정확도를 최소 15% 이상 개선~~해야~~ 하며, CatBoost를 기본 모델로 적용하고, **일부 특정** 점포의 경우 HistGradientBoostingRegressor(HGBR)를 병행하여 최적 성능을 확보~~해야~~ 한다. 모델 성능은 R^2 지표를 기준으로 평가~~해야~~ 하며, 목표는 0.80 이상~~이어야 한다~~ ~~이다~~. 모델은 정기적으로 재학습 및 성능 검증 과정을 수행~~해야~~ 한다.

3. 출력 요구사항

시스템은 예측 결과를 운영에 직접 활용할 수 있도록 CSV 리포트와 웹 대시보드 두 가지 형태로 제공~~해야~~ 한다. CSV 리포트는 예측치와 권장 발주량을 포함하여 주 1회 자동 생성·배포~~되어야 한다~~ ~~다~~ ~~된다~~. 웹 대시보드는 예측 결과를 실시간으로 조회·분석할 수 ~~있어야~~ ~~있도록~~ 하며, 점포별, 일

자별, 상품별 단위의 시각화 자료를 포함~~해야~~ 한다. 초기 검증 단계에서는 Streamlit 기반 프로토타입을 활용하여 CSV 다운로드, 예측치 vs 실제값 비교 그래프, ~~상품별-트렌드-조회~~ ~~상품별 외부 변수의 영향력 평가~~ 기능을 제공~~해야~~ 한다. 최종 단계에서는 JAVA 기반 대시보드를 협업사 운영 환경에 연동하여 서비스 수준으로 확장~~해야~~ 한다.

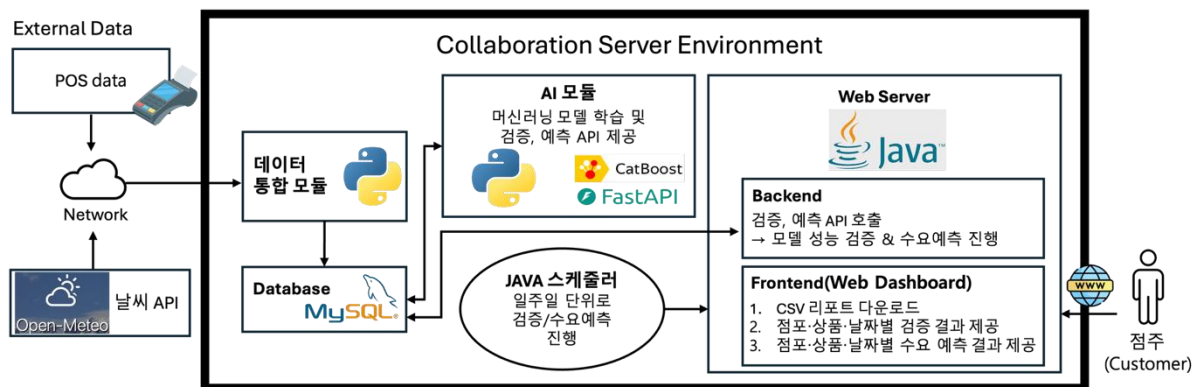
4. 사용자 요구사항

시스템은 최종 사용자가 예측 결과를 직관적으로 이해하고 활용할 수 있어야 한다. 제공되는 CSV 리포트는 간결하고 명확한 구조를 가져야 하며, 웹 대시보드 UI는 점포별 요약, 상품별 상세 분석 등 핵심 정보를 사용자 친화적으로 제공해야 한다. 이를 통해 점포 관리자~~는~~^가 예측 결과를 기반으로 합리적인 발주 결정을 내릴 수 ~~있어야 한다.~~ ~~있도록 한다.~~

3.2 전체 시스템 구성

~~본 프로젝트의 SW 전체 시스템 아래[그림 3.2.1]과 같이, 데이터 수집 및 통합 → 모델 학습 → 예측&검증 실행 → 결과 제공의 단계로 구성되며,~~

본 프로젝트의 SW 전체 아키텍처는 [그림 3.2]와 같이, 데이터 통합 모듈 → AI 모듈 → 웹 서버 모듈로 구성된다. 모든 핵심 모듈은 협업사 운영 서버 환경 내에서 상호작용하도록 통합 배포된다. 특히 중간 단계에서는 Streamlit 기반 프로토타입 대시보드(읽기 전용)를 추가하여 시각화 및 검증을 수행하고, 최종 단계에서는 **JAVA 기반 운영 대시보드로** 확장하는 구조를 지향한다.



[그림 3.2. SW 전체 아키텍처]

1. 데이터베이스 통합 모듈 (Python 환경): ~~시스템의 모든 정보는 중앙 DB에 저장된다.~~

POS system DB 및 Open-Meteo API와 연결되어 데이터를 수집·전처리하고 DB에 저장한다.

A. ~~저장 대상:~~ 원천 데이터(raw POS), 전처리 데이터, 학습 데이터셋, 학습된 모델 파일 (.cbm), 최종 예측 결과 csv 파일

B. ~~중간 단계 전용:~~ Streamlit 대시보드의 안전한 접속을 위해 읽기 전용 뷰(View)/스키마를 제공하며, 운영 모듈과는 권한을 분리하였다.

2. AI 모듈 (Python 환경) 데이터 수집 및 모델 학습 환경 (Python 기반)

- A. ~~데이터 통합 모듈: POS system DB 및 Open-Meteo API와 연결되어 데이터를 수집, 전처리하고 DB에 저장한다.~~
- B. ~~머신러닝 학습 모듈: Python 환경에서 CatBoost, HGBR 등의 머신러닝 모델을 학습, 튜닝한다. 학습 데이터는 DB에서 로드되며, 학습 완료된 모델(.cbm)은 다시 DB에 저장되어 운영 환경에서 재사용된다.~~
- C. 웹 서버 모듈로부터 요청이 있을 시, 머신러닝 모델의 성능 검증 및 수요예측을 진행한다.

3. 웹 서버 모듈 (Java 환경) ~~실질적 운영 및 서비스 환경(JAVA 기반)~~

- A. **JAVA 스케줄러(weekly automation):** 주 단위로 예측 엔진을 트리거하여 데이터 로딩, 모델 실행, 예측 결과 산출 및 검증, CSV 리포트 생성을 자동화한다.
- B. **JAVA Web Server:** Java-based Backend와 JSP-based Frontend로 구성된다.
 - ◆ Backend: ~~스케줄러에 의해 실행되는 예측 엔진을 포함하며, DB에서 예측 결과를 조회, 저장하고 CSV 리포트를 생성한다.~~ FastAPI 기반 서버로, 프론트엔드에서 발생하는 예측 및 검증 요청을 처리한다. 요청 시 AI 모듈을 호출하여 모델 학습 결과를 이용한 실시간 예측과 성능 검증을 수행한다. 처리 결과를 JSON 형태로 프론트엔드에 전달하며, CSV 리포트 생성 및 데이터베이스 저장 기능도 포함한다. 정기 자동 예측과 리포트 생성은 Java Scheduler 기반으로 수행된다.
 - ◆ Frontend: Backend를 통해 로드한 예측 및 검증 결과를 웹 대시보드 형태로 시각화하여 점주에게 제공한다.

4. 데이터베이스 (MySQL)

raw POS 데이터, 전처리 후 통합된 POS 데이터, 최종 예측 결과 csv 파일을 저장한다.

앞서 <2.5. 주요 기능 리스트>에서 언급한 주요 기능별 데이터 흐름을 SW 전체 아키텍처 관점에서 정리하면 다음과 같다.

주요기능	데이터 흐름
1. 데이터 전처리 자동화	POS 데이터+Open-Meteo 날씨 API → 데이터 통합 모듈 → 통합 데이터를 데이터베이스에 저장
2. 수요예측 엔진	데이터베이스에 저장된 통합 데이터 → CatBoost 또는 HGBR 모델 학습 → FastAPI 모델 배포
3. 결과 제공	FastAPI → CSV 변환 및 다운로드 / 데이터베이스에 CSV 저장
4. 웹 대시보드	FastAPI → Java 웹 대시보드 → 예측 결과 제공/모델 검증 결과 제공

~~본 프로젝트에서 제안하는 SW solution의 구조는 아래 [그림 3.2.2]와 같다.~~

~~이러한 운영 절차를 아키텍처 레벨에서 다시 정리하면, 본 프로젝트의 전체 시스템은 데이터 수집부터 예측 결과 제공까지 네 개의 계층(Data Source Layer, Model Training Layer, Prediction Layer, Presentation Layer)으로 나누어 설계되었다. 이 구조는 단순히 학습 모델 개발을 넘어, 실제 운영 가능한 서비스로 발전시키기 위한 기반을 제공한다.~~

~~1. Data Source Layer~~

~~데이터 수집 및 통합 단계로, 점포별 POS 데이터베이스와 외부 데이터를 통합한다. POS 데이터에는 일자별 상품명, 판매 수량, 매출액과 같은 핵심 지표가 포함되며, 이는 모델 학습의 기본 입력값으로 활용된다. 외부 데이터는 Open Meteo API를 통한 기온·강수량, 정부·지자체 제공 공휴일 정보, 통계청 민간 데이터를 활용한 유동인구 지표, 점포별 프로모션 이벤트 정보 등이 포함된다. 데이터는 매주 수집 갱신되며, 범주형 인코딩을 포함한 전처리를 거쳐 학습 가능한 형태로 정제된다. 초기에는 CSV 기반으로 운영되었으나, 협업사 (주)GN FOOD의 요구에 따라 DB 연동 방식으로 확장하여 안정적인 운영 환경을 구축하였다.~~

~~2. Model Training Layer~~

~~전처리된 데이터를 기반으로 파이썬 환경에서 수요 예측 모델을 학습한다. CatBoost, HGBR 등 다양한 머신러닝 모델을 적용해 성능을 비교·검증하고, 점포별 특성에 가장 적합한 모델을 최종 선택한다. 학습된 모델은 .cbm 파일 형태로 저장되어 운영 환경에서 재사용된다. 성능은 R^2 지표를 기준으로 평가하며(목표 ≥ 0.80), 기준에 미달할 경우 피쳐 엔지니어링·하이퍼파라미터 재조정을 거쳐 재학습한다. 초기 개발은 Google Colab GPU 환경에서 진행되었으며, 이후 협업사 서버와 연동 가능한 구조로 코드가 정리되었다.~~

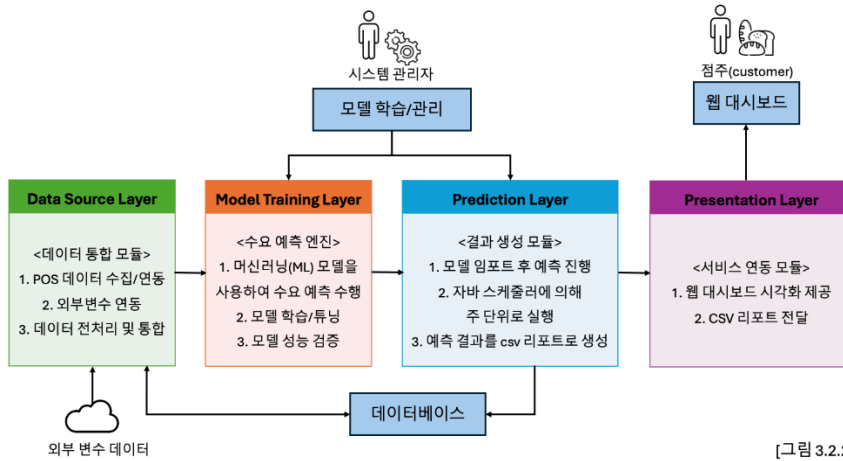
~~3. Prediction Layer~~

~~실제 운영 단계에서 자바 기반 예측 엔진이 Python에서 생성된 .cbm 모델을 불러와 예측을 수행한다. 예측 결과는 CSV 리포트로 생성되며, 자바 스케줄러에 의해 주 단위로 자동 실행된다. 산출된 지점 상품 단위 예측값은 DB에 저장되며, 단순 파일 출력에 그치지 않고 이후 웹 시스템에서 조회·활용할 수 있도록 구조화된다.~~

~~4. Presentation Layer~~

~~최종 사용자인 점포 운영자가 예측 결과를 직관적으로 확인·활용할 수 있도록 한다.~~

- ~~● CSV 리포트: 이미 운영 중이며, 주 1회 자동 생성되어 협업사에 제공된다. 리포트에는 지점별 상품별 예측치와 권장 발주량이 포함된다.~~
- ~~● 웹 대시보드: 현재 구현 중으로, 자바 기반 예측 모듈과 DB를 연동하여 지점별 필터링, 기간별 조회, 이벤트 관리 기능 등을 제공할 예정이다. 주요 화면은 지점 요약 대시보드, 상품별 상세 분석 화면, 이벤트 입력 화면으로 구성된다. 특히 이벤트 입력 화면은 관리자·자가 프로모션 유형과 기간을 직접 등록하면 해당 정보가 모델 입력값에 반영되도록 설계된다.~~

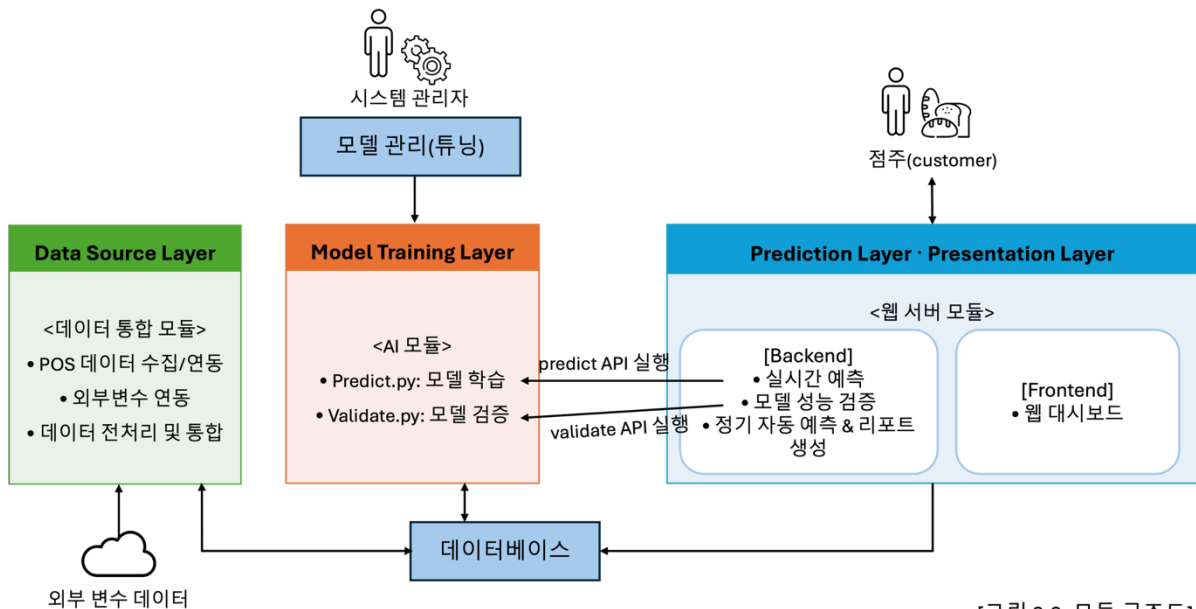


[그림 3.2.2]

종합적으로, 본 시스템은 데이터 소스 계층(Data Source Layer) → 모델 학습 계층(Model Training Layer) → 예측 실행 계층(Prediction Layer) → 시각화 계층(Presentation Layer)으로 나누어 설계되었다. 이는 단순 연구용 모델 개발 수준을 넘어, 협업사 환경에 실제 배포 가능한 **엔드투엔드 파이프라인**을 구현한다는 점에서 의미가 크다. 현재는 CSV 리포트 기반 결과 제공과 Streamlit 대시보드를 통한 중간 시각화 검증이 운영 중이며, 최종 단계에서는 JAVA 기반 대시보드와 협업사 시스템의 완전한 통합을 목표로 하고 있다.

3.3 주요 엔진 및 기능 설계

아래 [그림 3.3. 시스템 계층별 구성 및 데이터 흐름도]는 [그림 3.2. SW 전체 아키텍처]에서 모듈을 데이터 흐름에 따라 재구성한 것이다. 각 SW 모듈의 주요 구현 내용 및 방법은 다음과 같다.



[그림 3.3. 모듈 구조도]

1. 데이터 통합 모듈 (Data Source Layer)

데이터 수집 및 통합 단계로, 점포별 POS 데이터와 외부 변수 데이터를 결합하여 머신러닝 모델

학습에 활용 가능한 형태로 전처리한다.

POS 데이터는 일자별로 하나씩 존재하며, 각 일자별 데이터에는 상품명, 상품코드, 판매 수량, 매출액 등의 지표가 포함된다. 이 중 **상품명과 판매 수량**만을 모델 학습용 입력값으로 활용한다.

외부 변수 데이터는 날씨(최대·최소기온, 강수량), 공휴일 정보, 점포별 프로모션 이벤트 정보를 포함하며, 각각 다음과 같이 수집된다.

- 날씨 데이터: Open-Meteo API 활용
- 공휴일 데이터: 파이썬 내장 패키지 사용
- 점포별 프로모션 이벤트: 수작업으로 코드에 반영(하드코딩)

이후 POS 데이터와 외부 변수를 통합하고, 범주형 인코딩 등 전처리 과정을 거쳐 모델 학습이 가능한 구조의 통합 데이터셋으로 정제한다. 모든 데이터는 매주 수집·갱신되며, 결과는 데이터베이스에 저장되어 이후 머신러닝 모델 학습 및 검증에 사용된다.

2. AI 모듈 (Model Training Layer)

- **Predict.py**

데이터 통합 모듈에서 전처리된 데이터를 데이터베이스에서 불러오고, 이를 입력값으로 하여 파이썬 환경에서 수요 예측 모델(CatBoost, HGBR)을 학습한다.

- **점포별 모델 검증 및 선정 결과**

- **제주 애월점**

LGBM_Tweedie, XGB_RMSE, CatBoost, HGBR 등 다양한 모델을 학습·검증한 결과, HGBR 모델의 예측 정확도가 가장 높게 나타났다. 8월 중순 1차 모델로 확정하였으며, 이후 8주간의 검증 결과 R^2 평균 0.8 이상을 안정적으로 유지함을 확인하였다. 이에 따라 10월 중순 최종 모델로 확정하였다.

- **연남점**

동일한 모델군을 적용한 결과, CatBoost 모델의 예측 정확도가 가장 높게 나타났다. 8월 말 1차 확정 후 4주간 검증 결과, R^2 평균 0.6 이하로 다소 낮은 성능을 보였다. 이에 공휴일 변수를 0~1 사이로 스케일링하여 추가 튜닝을 수행하여, 10월 중순 재학습 후 최종 모델로 확정하였다.

- **수원 타임빌라스점**

연남점과 유사한 도시형 점포로, 동일한 CatBoost 모델 소스코드를 기반으로 학습하였다. 점포 입지 특성상 수요 변동성이 낮은 점을 반영하여, 8월 말 1차 모델 확정 후 8주간의 검증 결과 R^2 평균 0.8 이상으로 유지되었고, 10월 중순 최종 모델로 확정하였다.

- **부산 광안리점**

휴가철·계절 요인 등으로 인한 급격한 수요 변동성을 반영하기 위해, 제주 애월점 모델을 참고하여 HGBR 모델을 적용하였다. 초기에는 $R^2 \geq 0.8$ 수준의 예

측 성능을 확보했으나, 이후 5주간의 검증에서 예측 정확도 하락이 관찰되었다. 이에 CatBoost모델로 변경 후 추가 튜닝을 진행하여 10월 중순 최종 모델을 확정하였다.

- **Validate.py**

데이터베이스에 저장된 실제 POS 판매 데이터와 수요 예측 결과 데이터를 비교하여, CatBoost, HGBR 모델 성능을 R^2 지표(목표 ≥ 0.80)를 기준으로 평가한다. 만약 성능이 기준에 미달할 경우 시스템 관리자는 피쳐 엔지니어링 및 하이퍼파라미터 튜닝을 다시 수행한 후 모델을 재학습한다.

모델 학습 및 검증 파이프라인은 Python 스크립트 형태로 자동화되어 있으며, 초기 모델 개발은 Google Colab GPU 환경의 Jupyter Notebook(.ipynb)으로 구현되었다. 이후 협업사 서버와의 연동을 위해 Python 스크립트(.py) 기반 구조로 변환 및 정리되었다.

Predict.py, Valdiate.py는 버전 관리되어 FastAPI 서버에서 실시간 예측 및 검증 요청 시 불러올 수 있도록 구성하였다.

3. 웹 서버 모듈 (Prediction Layer • Presentation Layer)

- ① **(사용자 요청 시) 실시간 예측 (FastAPI 기반)**

사용자가 웹 페이지에서 기준일자와 점포를 설정하고 예측 버튼을 클릭하면, FastAPI 서버의 predict API가 실행된다. 이 API는 AI 모듈 내 predict.py를 호출하여, 데이터베이스에 저장된 최신 전처리 데이터를 모델 입력값으로 불러오고, 향후 7일간의 상품별 판매 수량을 실시간으로 예측한다. 예측 결과는 JSON 형식으로 프론트엔드에 전달되어 대시보드에 시각화되며, 동시에 CSV 리포트 형태로 변환되어 사용자가 웹 페이지에서 직접 다운로드할 수 있다.

- ② **(사용자 요청 시) 모델 성능 검증 (FastAPI 기반)**

사용자가 대시보드에서 특정 점포와 기간을 선택하여 검증 요청을 보낼 경우, FastAPI 서버의 validate API가 실행되어 AI 모듈의 validate.py를 호출한다. 이 스크립트는 해당 구간의 실제 판매 수량 데이터와 예측값을 비교하여 R^2 지표를 계산하고, 결과를 JSON 형태로 응답한다. 응답된 데이터는 프론트엔드에서 표와 그래프 형태로 시각화되어, 사용자가 모델의 지난 예측 정확도를 한눈에 파악할 수 있도록 한다.

- ③ **정기 자동 예측 및 리포트 생성 (Java Scheduler 기반)**

실시간 요청 외에도, 자바 스케줄러에 의해 매주 월요일 새벽 3시에 자동으로 예측 및 리포트 생성 작업이 수행된다.

스케줄러는 모든 점포의 최신 데이터를 불러와 일괄 예측을 수행하고, 생성된 결과 CSV 리포트를 데이터베이스에 저장한다. 리포트에는 점포별·일자별·상품별 예측치와 권장 발주량이 포함된다. 이때, 품질비용이 폐기비용보다 크다는 점을 고려하여 수요 예측치를 올림(ceil) 처리하여 협업사에게 주기적으로 제공한다. 이 리포트는 협업사에 제공하는 것 외에, 검증 API 실행 시 비교 기준 데이터로도 활용된다.

- ④ **웹 대시보드(Frontend)**

최종 사용자인 점포 운영자가 예측 및 검증 결과를 직관적으로 확인할 수 있도록 Java 기반 웹 대시보드를 구현하였다. 대시보드는 점포별 검증 결과 화면, 수요 예측 결과 화면으로 구성되며, 필터링(점포, 일자, 상품)과 수요 예측 CSV 리포트 다운로드 기능을 제공한다.

현재 주요 화면 구현은 완료된 상태이며, 향후 사용자 편의를 위해 시각화 개선을 진행할 예정이다.

3.4 주요 기능의 구현

본 프로젝트에서는 데이터 전처리 자동화, 수요 예측, 결과 제공, 웹 대시보드의 4가지 주요 기능을 구현하였으며, 이를 위해 데이터 통합 모듈, AI 모듈, 웹 서버 모듈의 3가지 SW 모듈을 구성하였다. 각 기능과 모듈의 연관성 및 구현 방식은 다음과 같다.

1. 데이터 전처리 자동화

본 기능의 구현을 위해 **데이터 통합 모듈(Data Source Layer)**을 도입하였다.

데이터 통합 모듈은 파이썬 환경에서 점포별 POS 데이터와 외부 변수 데이터를 수집 및 통합하여 머신러닝 모델 학습에 활용 가능한 형태로 전처리한다. 구체적으로, POS 데이터는 상품명, 상품코드, 판매 수량, 매출액 등의 정보를 포함하나, 모델 학습용으로는 상품명과 판매 수량만을 추출하였다. 외부 변수 데이터는 날씨, 공휴일, 점포별 프로모션 이벤트를 포함하며, 날씨 데이터는 Open-Meteo API, 공휴일 정보는 파이썬 내장 패키지를 활용하고, 프로모션 이벤트는 하드코딩 방식으로 반영하였다. 통합 후, 범주형 인코딩 및 기타 전처리 과정을 거쳐 모델 학습용 데이터셋을 생성하며, 결과는 데이터베이스에 저장되어 이후 AI 모듈에서 재사용된다.

2. 수요 예측 엔진

본 기능의 구현을 위해 **AI 모듈(Model Training Layer)**을 활용하였다.

AI 모듈 내 Predict.py에서는 데이터 통합 모듈에서 전처리된 데이터를 불러와 **CatBoost와 HGBR 모델을 학습**하였다. 다양한 머신러닝 모델을 적용해 성능을 비교 검증한 결과, 점포별 특성에 가장 적합한 모델을 최종 선택하였다.

Validate.py는 학습된 모델을 이용하여 실제 POS 판매 데이터와 예측 결과를 비교하고, **R² 지표 (목표 ≥ 0.80)** 기준으로 모델 성능을 평가한다. 성능이 기준에 미달할 경우, 시스템 관리자가 피쳐 엔지니어링과 하이퍼파라미터 튜닝을 수행한 후 모델을 재학습하도록 되어 있다.

Predict.py와 Validate.py는 Python 기반으로 구현되었으며, FastAPI 서버를 통해 사용자의 실시간 예측 및 검증 요청 시 웹 서버 모듈에서 호출될 수 있도록 연동하였다.

3. (예측) 결과 제공 및 웹 대시보드

본 기능의 구현을 위해 **AI 모듈(Model Training Layer)** 및 **웹 서버 모듈(Prediction Layer · Presentation Layer)**을 활용하였다.

사용자가 웹 대시보드에서 기준일자와 점포를 선택하면, 웹 서버 모듈 내 백엔드의 predict API가 AI 모듈의 Predict.py를 호출하여 향후 7일간 상품별 판매 수량을 실시간 예측한다. 예측 결과는 JSON 형식으로 웹 서버 모듈 내 프론트엔드에 전달되어 대시보드에 시각화되며, 동시에 CSV 리포트 형태로 다운로드할 수 있도록 제공된다.

사용자의 모델 성능 검증 요청 시에는 웹 서버 모듈 내 백엔드의 validate API가 AI 모듈의 Validate.py를 호출하여 R^2 지표를 계산하고, 결과를 프론트엔드로 전달하여 그래프 및 표로 시각화한다.

정기 자동 예측 및 리포트 생성 기능은 Java Scheduler 기반으로 수행되며, 매주 월요일 새벽 3시에 모든 점포 데이터를 불러와 일괄 예측을 수행하고, 결과 리포트를 데이터베이스에 저장한 후 협업사에 전달한다.

최종 사용자인 점포 운영자는 웹 서버 모듈을 통해 Java 기반 웹 대시보드에서 예측 및 검증 결과를 직관적으로 확인할 수 있으며, 점포, 기간, 상품별 필터링과 CSV 다운로드 기능을 활용할 수 있다.

[1] 오피니언뉴스, 「AI가 만든 빵, CJ푸드빌 뚜레쥬르 일부 매장서 판매」, 2023.
<https://www.opinionnews.co.kr/news/articleView.html?idxno=121958>

[2] CJ AI센터 뉴스, 「CJ푸드빌, AI 기반 빵 수요예측 모델 도입」, 2023.
<https://aicenter.cj.net/news/news/17>

[3] 그린경제, 「SPC·롯데제과, 빅데이터로 유통 효율화 시도」, 2023.
<https://www.greened.kr/news/articleView.html?idxno=296381>

[4] Blue Yonder, *Demand Forecasting Solutions*, 공식 웹사이트, 2025.
<https://blueyonder.com/solutions/supply-chain-planning/demand-forecasting>