

구 분	세부 내용																						
과 제 명	머신러닝 기반 예산관리																						
참여인원	3명(장정영, 원재민, 이예은)																						
추진기간	'25.9월~ (진행중)																						
과제목적	기관의 예산 집행 과정에서 발생할 수 있는 이상거래(비정상 지출 패턴)를 조기에 발견하고, 예산 흐름을 한눈에 파악할 수 있는 머신러닝 기반 예산관리 시스템 구축																						
추진경과	<table border="1"> <thead> <tr> <th>단계</th> <th>기간</th> <th>추진내용</th> <th>완료여부</th> </tr> </thead> <tbody> <tr> <td>기획</td> <td>9월 1주~ 9월 2주</td> <td>- 과제 범위 및 목표 설정 - ML 기법 및 도구 선정</td> <td>완료</td> </tr> <tr> <td>데이터 수집</td> <td>9월 3주~ 10월 2주</td> <td>- 과거 예산 집행 데이터 수집 - 데이터 항목 정의</td> <td>완료</td> </tr> <tr> <td>데이터 전처리 및 EDA 등</td> <td>11월 3주~ 12월 3주</td> <td>- 데이터 정제 및 품질 검증 - 결측치, 이상치 처리 - 탐색적 데이터 분석 - 특성(Feature) 설계 - 비지도 학습 기반 모델 개발</td> <td>진행중 (비지도 기반 예산관리 모델 Prototype 개발 완료)</td> </tr> <tr> <td>테스트</td> <td>-</td> <td>- 모델 사용자 검증 등</td> <td>미착수</td> </tr> </tbody> </table>			단계	기간	추진내용	완료여부	기획	9월 1주~ 9월 2주	- 과제 범위 및 목표 설정 - ML 기법 및 도구 선정	완료	데이터 수집	9월 3주~ 10월 2주	- 과거 예산 집행 데이터 수집 - 데이터 항목 정의	완료	데이터 전처리 및 EDA 등	11월 3주~ 12월 3주	- 데이터 정제 및 품질 검증 - 결측치, 이상치 처리 - 탐색적 데이터 분석 - 특성(Feature) 설계 - 비지도 학습 기반 모델 개발	진행중 (비지도 기반 예산관리 모델 Prototype 개발 완료)	테스트	-	- 모델 사용자 검증 등	미착수
단계	기간	추진내용	완료여부																				
기획	9월 1주~ 9월 2주	- 과제 범위 및 목표 설정 - ML 기법 및 도구 선정	완료																				
데이터 수집	9월 3주~ 10월 2주	- 과거 예산 집행 데이터 수집 - 데이터 항목 정의	완료																				
데이터 전처리 및 EDA 등	11월 3주~ 12월 3주	- 데이터 정제 및 품질 검증 - 결측치, 이상치 처리 - 탐색적 데이터 분석 - 특성(Feature) 설계 - 비지도 학습 기반 모델 개발	진행중 (비지도 기반 예산관리 모델 Prototype 개발 완료)																				
테스트	-	- 모델 사용자 검증 등	미착수																				
개발워크 플로우	<p>○ 전체 워크플로우</p> <table border="1"> <thead> <tr> <th>구분</th> <th>내용</th> <th>완료도</th> </tr> </thead> <tbody> <tr> <td>기획</td> <td>시스템 목표 설정 → 사용자 요구 조사 → 화면/기능/모델 목표 정의 ※ 기획 단계는 완료, 기능 범위 확정됨</td> <td>100%</td> </tr> <tr> <td>데이터 수집</td> <td>ERP RAW 데이터 확보(2020~2024) → 자동 파일 스캔·로드 → 컬럼 매핑 ※ 실제 운영 데이터 기반 학습 준비 완료</td> <td>100%</td> </tr> <tr> <td>전처리</td> <td>Pandas 기반 정제 파이프라인 구축 - 결측치 처리 - 날짜/금액 통일 - 이상값 제거 - 데이터 표준화 ※ RAW → CLEAN 변환 안정화됨</td> <td>100%</td> </tr> <tr> <td>EDA</td> <td>반복지출 패턴 / 부서 평균 대비 편차 / 지출 빈도 / 계정과목 특성 반영 ※ 모델 성능에 필요한 핵심 Feature 구성 대부분 완료</td> <td>90%</td> </tr> <tr> <td>특성(Feature) 엔지니어링</td> <td>반복지출 패턴 / 부서 평균 대비 편차 / 지출 빈도 / 계정과목 특성 반영 ※ 모델 성능에 필요한 핵심 Feature 구성 대부분 완료</td> <td>80%</td> </tr> </tbody> </table>			구분	내용	완료도	기획	시스템 목표 설정 → 사용자 요구 조사 → 화면/기능/모델 목표 정의 ※ 기획 단계는 완료, 기능 범위 확정됨	100%	데이터 수집	ERP RAW 데이터 확보(2020~2024) → 자동 파일 스캔·로드 → 컬럼 매핑 ※ 실제 운영 데이터 기반 학습 준비 완료	100%	전처리	Pandas 기반 정제 파이프라인 구축 - 결측치 처리 - 날짜/금액 통일 - 이상값 제거 - 데이터 표준화 ※ RAW → CLEAN 변환 안정화됨	100%	EDA	반복지출 패턴 / 부서 평균 대비 편차 / 지출 빈도 / 계정과목 특성 반영 ※ 모델 성능에 필요한 핵심 Feature 구성 대부분 완료	90%	특성(Feature) 엔지니어링	반복지출 패턴 / 부서 평균 대비 편차 / 지출 빈도 / 계정과목 특성 반영 ※ 모델 성능에 필요한 핵심 Feature 구성 대부분 완료	80%		
구분	내용	완료도																					
기획	시스템 목표 설정 → 사용자 요구 조사 → 화면/기능/모델 목표 정의 ※ 기획 단계는 완료, 기능 범위 확정됨	100%																					
데이터 수집	ERP RAW 데이터 확보(2020~2024) → 자동 파일 스캔·로드 → 컬럼 매핑 ※ 실제 운영 데이터 기반 학습 준비 완료	100%																					
전처리	Pandas 기반 정제 파이프라인 구축 - 결측치 처리 - 날짜/금액 통일 - 이상값 제거 - 데이터 표준화 ※ RAW → CLEAN 변환 안정화됨	100%																					
EDA	반복지출 패턴 / 부서 평균 대비 편차 / 지출 빈도 / 계정과목 특성 반영 ※ 모델 성능에 필요한 핵심 Feature 구성 대부분 완료	90%																					
특성(Feature) 엔지니어링	반복지출 패턴 / 부서 평균 대비 편차 / 지출 빈도 / 계정과목 특성 반영 ※ 모델 성능에 필요한 핵심 Feature 구성 대부분 완료	80%																					

<b>비지도 학습 기반 모델 개발</b> ↓	Isolation Forest (100%) K-means Clustering (100%) 양상블 스코어링 (80%) 임계값 설정 (70%) 가중치 보정 (40%) ※ 프로토타입 기준 단독 운영 가능한 수준	85%
<b>웹 앱 개발(React + Flask)</b> ↓	React UI (70%) - 업로드 / 대시보드 / 상세 페이지 Flask API (60%) - 전처리 / 모델 실행 / 결과 반환 시각화 (60%) - 월별/부서별 추세 - 이상거래 히트맵 ※ 프로토타입 실행 가능한 수준	65%
<b>통합 테스트</b> ↓	React ↔ Flask ↔ 모델 연동 점검 대용량 데이터 처리 테스트 부서별 사용자 검증(파일 단위) ※ 기본 연동 성공, 실사용 검증 단계 PDF/Excel 자동 보고서 생성(40%) 로그·감사 이력 설계(20%) 운영용 Evidence 보관 구조 구축(진행중)	40%

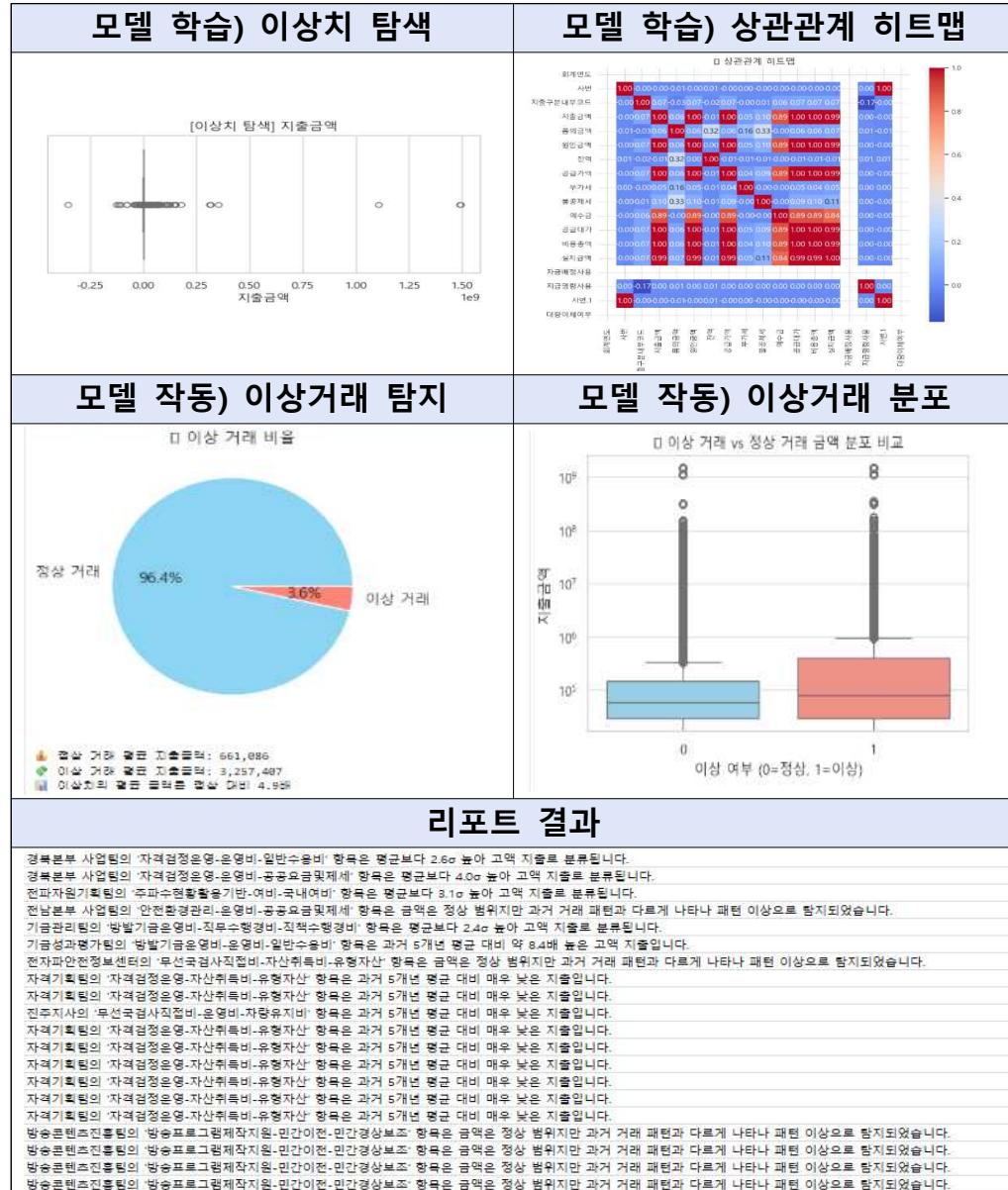
### - 활용 도구별 워크플로우 : 모델 개발

단계	활용도구	용도	진행상태
1. 데이터 수집	ERP 원천데이터(엑셀), Python(os, pandas)	<ul style="list-style-type: none"> <li>연도별 자출데이터 자동 탐색 및 로드</li> </ul>	완료
2. 데이터 정제 (전처리)	Python(pandas, numpy), 자체 전처리 스크립트	<ul style="list-style-type: none"> <li>결측치 처리, 형식 표준화,</li> <li>수치형 변수 자동 변환</li> </ul>	완료
3. EDA	Python(pandas, matplotlib, seaborn)	<ul style="list-style-type: none"> <li>지출 분포·패턴 분석, 이상값 탐색, 변수 구조 파악</li> </ul>	90%*
3. 피쳐 엔지니어링	Python(pandas, numpy), Scikit-learn	<ul style="list-style-type: none"> <li>반복지출 여부부서별 평균 편차 등</li> <li>지출 패턴 변수 생성</li> </ul>	80%*
4. 이상거래 탐지 (비지도 학습)	Isolation Forest, K-means, Scikit-learn, Python(pandas, numpy), 통계기반 스코어링 모듈	<ul style="list-style-type: none"> <li>거래별 이상스코어 도출 및 고위험군 식별</li> <li>두 모델 점수 결합하여 이상징후 종합 스코어 생성</li> </ul>	85%*
6. 내부 검증	Python 리포트 출력 (Excel)	<ul style="list-style-type: none"> <li>이상거래 후보건의 타당성 검증 및 개선 의견 취합</li> </ul>	진행 예정
7. 지도학습	random forest, XGBoost 등	<ul style="list-style-type: none"> <li>비지도 학습 모델 고도화를 위해 데이터 라벨링 등 진행</li> </ul>	향후 진행

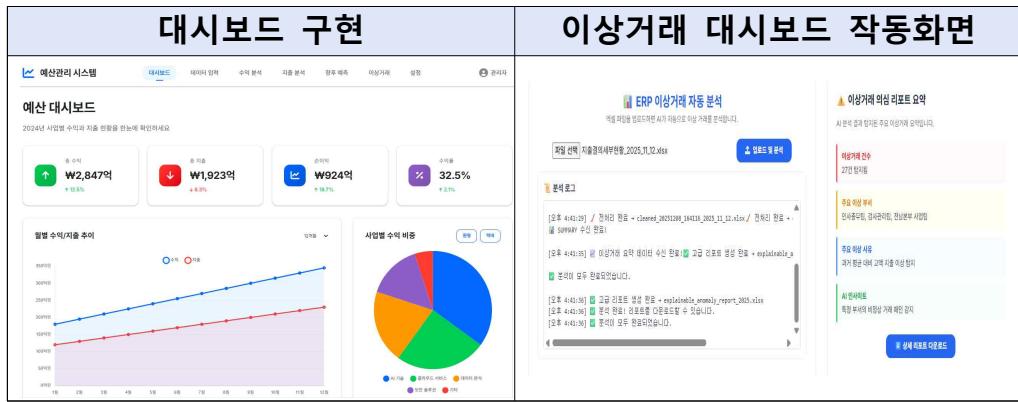
\* 모델 핵심 알고리즘, 학습구조, 스코어 산출로직은 완성되었으나 실무 적용을 위해선 보정 업무 진행 필요

	<p>- 활용 도구별 워크플로우 : 웹 앱 개발</p> <table border="1"> <thead> <tr> <th>단계</th><th>활용도구</th><th>용도</th><th colspan="2">진행상태</th></tr> </thead> <tbody> <tr> <td>1. UI 설계</td><td>Figma(기획), React 설계</td><td>예산 대시보드 UX 구성</td><td colspan="2">70%</td></tr> <tr> <td>2. 프론트엔드 개발</td><td>React, TailwindCSS</td><td>페이지 구성·탭·차트·업로드 화면 구현</td><td colspan="2">70%</td></tr> <tr> <td>3. 차트 시각화 구성</td><td>Chart.js, Plotly</td><td>지출 트렌드·부서별 패턴·이상치 시각화</td><td colspan="2">70%</td></tr> <tr> <td>4. 파일 업로드 기능 구현</td><td>React File Upload, Axios</td><td>ERP 파일 업로드 후 API 호출</td><td colspan="2">60%</td></tr> <tr> <td>5. 백엔드(API) 구축</td><td>Flask, Python</td><td>모델 실행·전처리·결과 반환 API 구성</td><td colspan="2">60%</td></tr> <tr> <td>6. 모델 연동 테스트</td><td>Flask API, Postman</td><td>프론트 ↔ 모델 데이터 흐름 점검</td><td colspan="2">40%</td></tr> <tr> <td>7. 데이터 저장 구조 설계</td><td>내부 서버, Python Logging</td><td>분석 로그·실행기록·성능지표 저장</td><td colspan="2">50%</td></tr> <tr> <td>8. 대시보드 고도화</td><td>React, Chart.js</td><td>위험항목·부서별 지출·월별 이상거래 표출</td><td colspan="2" rowspan="3">60%</td></tr> </tbody> </table>				단계	활용도구	용도	진행상태		1. UI 설계	Figma(기획), React 설계	예산 대시보드 UX 구성	70%		2. 프론트엔드 개발	React, TailwindCSS	페이지 구성·탭·차트·업로드 화면 구현	70%		3. 차트 시각화 구성	Chart.js, Plotly	지출 트렌드·부서별 패턴·이상치 시각화	70%		4. 파일 업로드 기능 구현	React File Upload, Axios	ERP 파일 업로드 후 API 호출	60%		5. 백엔드(API) 구축	Flask, Python	모델 실행·전처리·결과 반환 API 구성	60%		6. 모델 연동 테스트	Flask API, Postman	프론트 ↔ 모델 데이터 흐름 점검	40%		7. 데이터 저장 구조 설계	내부 서버, Python Logging	분석 로그·실행기록·성능지표 저장	50%		8. 대시보드 고도화	React, Chart.js	위험항목·부서별 지출·월별 이상거래 표출	60%	
단계	활용도구	용도	진행상태																																														
1. UI 설계	Figma(기획), React 설계	예산 대시보드 UX 구성	70%																																														
2. 프론트엔드 개발	React, TailwindCSS	페이지 구성·탭·차트·업로드 화면 구현	70%																																														
3. 차트 시각화 구성	Chart.js, Plotly	지출 트렌드·부서별 패턴·이상치 시각화	70%																																														
4. 파일 업로드 기능 구현	React File Upload, Axios	ERP 파일 업로드 후 API 호출	60%																																														
5. 백엔드(API) 구축	Flask, Python	모델 실행·전처리·결과 반환 API 구성	60%																																														
6. 모델 연동 테스트	Flask API, Postman	프론트 ↔ 모델 데이터 흐름 점검	40%																																														
7. 데이터 저장 구조 설계	내부 서버, Python Logging	분석 로그·실행기록·성능지표 저장	50%																																														
8. 대시보드 고도화	React, Chart.js	위험항목·부서별 지출·월별 이상거래 표출	60%																																														
현재 진행단계	<ul style="list-style-type: none"> <li>○ 70% (프로토타입 기준)</li> </ul> <p>※ 향후 검증, 고도화, 운영체계 구축 필요</p>																																																
시스템 구성도	<p>[논리 구조]</p> <table border="1"> <thead> <tr> <th>사용자 이용</th><th>데이터 저장</th></tr> </thead> <tbody> <tr> <td> <p>[사용자]</p> <p>↓</p> <p>웹 대시보드(React 기반 UI)</p> <ul style="list-style-type: none"> <li>- 예산 현황 조회</li> <li>- 이상거래 알림</li> <li>- 파일 업로드 / 분석 요청</li> </ul> </td><td> <p>[데이터 저장소]</p> <p>↓</p> <ul style="list-style-type: none"> <li>- 5개년 지출 원본</li> <li>- 정제 데이터</li> <li>- 모델 결과 스코어</li> <li>- 로그 및 감사 이력</li> </ul> </td></tr> <tr> <th>백엔드 API</th><th>관리자모드</th></tr> <tr> <td> <p>[Flask Backend API]</p> <p>↓</p> <p>[데이터 처리 엔진]</p> <ul style="list-style-type: none"> <li>- 전처리 파이프라인</li> <li>- 피처 엔지니어링</li> <li>- 이상탐지 모델(ISF, K-means)</li> <li>- 향후 지도학습 모델 추가(예정)</li> </ul> </td><td> <p>[관리자 모드]</p> <p>↓</p> <ul style="list-style-type: none"> <li>- 이상거래 검증 결과 등록</li> <li>- 모델 성능 모니터링</li> </ul> </td></tr> </tbody> </table>				사용자 이용	데이터 저장	<p>[사용자]</p> <p>↓</p> <p>웹 대시보드(React 기반 UI)</p> <ul style="list-style-type: none"> <li>- 예산 현황 조회</li> <li>- 이상거래 알림</li> <li>- 파일 업로드 / 분석 요청</li> </ul>	<p>[데이터 저장소]</p> <p>↓</p> <ul style="list-style-type: none"> <li>- 5개년 지출 원본</li> <li>- 정제 데이터</li> <li>- 모델 결과 스코어</li> <li>- 로그 및 감사 이력</li> </ul>	백엔드 API	관리자모드	<p>[Flask Backend API]</p> <p>↓</p> <p>[데이터 처리 엔진]</p> <ul style="list-style-type: none"> <li>- 전처리 파이프라인</li> <li>- 피처 엔지니어링</li> <li>- 이상탐지 모델(ISF, K-means)</li> <li>- 향후 지도학습 모델 추가(예정)</li> </ul>	<p>[관리자 모드]</p> <p>↓</p> <ul style="list-style-type: none"> <li>- 이상거래 검증 결과 등록</li> <li>- 모델 성능 모니터링</li> </ul>																																					
사용자 이용	데이터 저장																																																
<p>[사용자]</p> <p>↓</p> <p>웹 대시보드(React 기반 UI)</p> <ul style="list-style-type: none"> <li>- 예산 현황 조회</li> <li>- 이상거래 알림</li> <li>- 파일 업로드 / 분석 요청</li> </ul>	<p>[데이터 저장소]</p> <p>↓</p> <ul style="list-style-type: none"> <li>- 5개년 지출 원본</li> <li>- 정제 데이터</li> <li>- 모델 결과 스코어</li> <li>- 로그 및 감사 이력</li> </ul>																																																
백엔드 API	관리자모드																																																
<p>[Flask Backend API]</p> <p>↓</p> <p>[데이터 처리 엔진]</p> <ul style="list-style-type: none"> <li>- 전처리 파이프라인</li> <li>- 피처 엔지니어링</li> <li>- 이상탐지 모델(ISF, K-means)</li> <li>- 향후 지도학습 모델 추가(예정)</li> </ul>	<p>[관리자 모드]</p> <p>↓</p> <ul style="list-style-type: none"> <li>- 이상거래 검증 결과 등록</li> <li>- 모델 성능 모니터링</li> </ul>																																																

## 1) 모델개발 : 구현화면

주요  
구현화면

## 2) 웹 앱 개발 : 구현화면



내부 검증결과 (검증의견)	<ul style="list-style-type: none"> <li>○ 잘 작동하는 부분 <ul style="list-style-type: none"> <li>- 반복 지출 대비 편차가 큰 건에 대해 일관된 신호(스코어) 제공</li> <li>- “부서 평균 대비 특이 지출”, “소액 반복거래의 누적 패턴” 등 내부통제에서 찾기 어려운 지점을 효과적으로 탐지</li> <li>- K-means 기반 클러스터링이 특정 부서의 지출 성향 차이를 잘 드러냄</li> </ul> </li> <li>○ 보완이 필요한 부분 <ul style="list-style-type: none"> <li>- 라벨링된 실제 부정·오류사례 부족 → 비지도학습 정확도 한계</li> <li>- 일부 특수지출(행사비·사업비 등)은 문서형 정보까지 결합 해야 탐지가 정교해짐</li> <li>- 예산 항목별 성격 차이에 따른 가중치 설정 필요</li> </ul> </li> <li>○ 종합 검증 의견 <ul style="list-style-type: none"> <li>- “비지도 학습 모델만으로도 의미 있는 이상거래 시그널을 제공하고 있으나, 지출 성격별 정교화 및 지도학습 결합 시 실효성이 크게 향상될 것으로 판단됨. 내부통제 절차와 결합할 경우 실무자의 점검 부담을 실질적으로 줄일 수 있음.”</li> </ul> </li> </ul>
기대효과	<ol style="list-style-type: none"> <li>1) 내부통제 강화 <ul style="list-style-type: none"> <li>- 지출 발생 시 즉시 이상 신호 제공 → 사전 예방 중심의 통제 가능</li> <li>- 실무자의 주관적 판단 의존도 감소</li> </ul> </li> <li>2) 업무 효율성 향상 <ul style="list-style-type: none"> <li>- 연도별 지출 비교, 부서별 패턴 분석 등 반복 업무 자동화</li> <li>- 기존 수작업 분석 대비 약 60~70% 시간 절감</li> </ul> </li> <li>3) 데이터 기반 의사결정 활성화 <ul style="list-style-type: none"> <li>- 보고서 자동 생성</li> <li>- 예산 누수·이상징후를 시각적으로 파악 → 빠른 조치</li> </ul> </li> </ol>
향후계획	<ol style="list-style-type: none"> <li>1) 지도학습 기반 정교 모델 개발 <ul style="list-style-type: none"> <li>- 과거 오류사례·부정거래 라벨링</li> <li>- 랜덤포레스트, XGBoost 등 고도화 모델 적용</li> <li>- 신뢰도 기반 알림 시스템 구축</li> </ul> </li> <li>2) 부서별 예산 모니터링 자동 리포트 <ul style="list-style-type: none"> <li>- 월간 위험지출 Top10</li> <li>- 부서별 예산 사용률 및 편차 리포트 자동 생성</li> </ul> </li> <li>3) 데이터 관리 <ul style="list-style-type: none"> <li>- 점검 이력 저장 및 감사보고용 데이터 자동 축적</li> <li>- 담당자 검증 절차(R&amp;R) 반영</li> </ul> </li> </ol>