
2025 SW중심대학 디지털 경진대회

SW 부문

아이디어 기획서

팀명	DMS
대학명	창원대학교
서비스명(프로젝트명)	Captain AI

팀명	DMS
대학명	창원대학교
팀원(학과)	박진성(컴퓨터공학), 김성연(컴퓨터공학), 김남규(컴퓨터공학), 윤은옥(컴퓨터공학), 김민재(컴퓨터공학)
서비스명 (프로젝트명)	Captain AI
한줄소개 (서비스 요약)	기상 데이터 기반 Go-Around 예측 모델을 통해 항공 안전 위험을 줄이기 위한 서비스
주요 활용 데이터 (제공기관명)	기상 관련 정보(한국 기상청API) 항공 데이터(flightrader24)

개발동기
및
서비스
제안 배경

- ✓ 기상 데이터 기반 Go-Around 예측 모델의 부재로 인한 항공 안전 위험을 줄이기 위해 서비스 기획
- ✓ Generative AI Chatbot을 통해 조종사 및 관제사에게 실시간 예측과 대응 가이드 제공

기대효과

- ✓ Go-Around 가능성 사전 예측으로 항공사고 위험 감소
- ✓ Generative AI Chatbot을 통한 실시간 안내로 조종사와 관제사의 대응 효율성 향상

개발동기

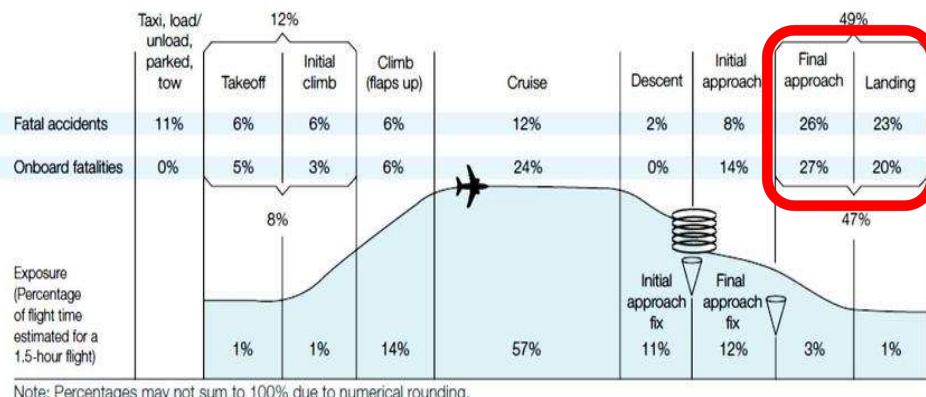


FIGURE 1 Percentage of fatal accidents and onboard fatalities (Boeing, 2015).

Donavalli, Bharath, "IMPACT OF WEATHER FACTORS ON GO-AROUND OCCURRENCE" (2016). Civil Engineering Theses. 445.

항공 단계별 사고 발생률



https://weekly.khan.co.kr/khnm.html?mode=view&art_id=202501060600031&code=115
주간경향, 원인 규명에 시간 필요.."조용히 기다려야", 2025.01.13, 김찬호 기자

항공 여객 수송 현황

착륙 단계에서 전체 항공사고의 48%가 발생하며 과거보다 항공기 사고 빈도는 분명히 줄어들고 있으나, 최근 '제주항공 여객기 참사'와 같은 사고가 발생한다면 탑승객의 생존 확률은 높지 않음 ⇒ 사고 빈도를 최대한 줄일 수 있도록 도와주는 AI 프로젝트 계획

개발동기



Federal Aviation
Administration

- 착륙 복행은 안전한 착륙이 어렵다고 조종사 또는 관제사가 주관적 판단으로 즉시 수행되는 절차
- 현재 조종사 및 관제사의 경험과 현장 상황에 의존하여 시행되고 있어, 어려움 존재
- 이를 돋기 위한 서비스를 개발함으로써 조종사의 부담을 줄이고자 함

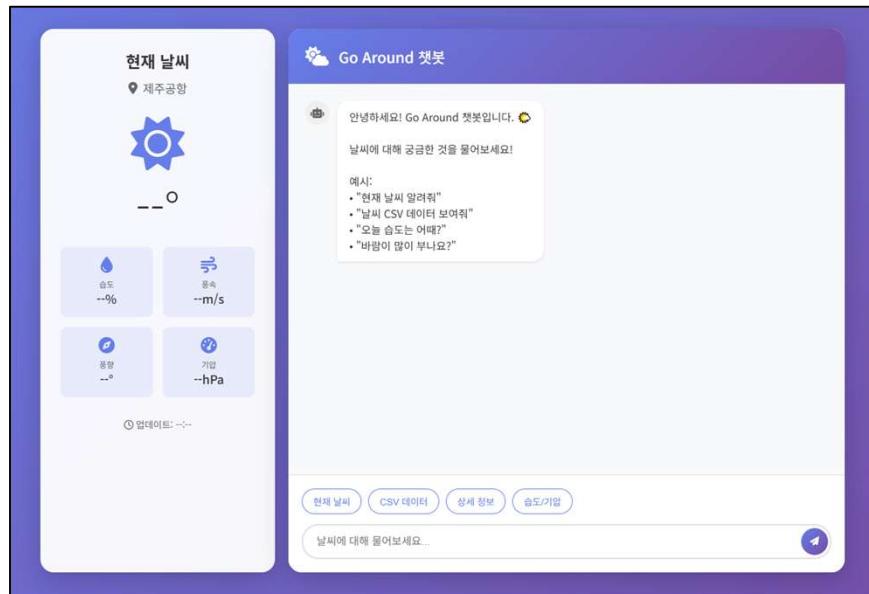
① ICAO Annex 6 (Operation of Aircraft) 4-6-4 Approach and Landing

"If an approach cannot be continued safely, a missed approach shall be executed"
“안전하게 접근을 계속할 수 없다고 판단 시 복행을 조종사가 즉시 수행해야 함”

② FAA ATC Handbook (FAA JO 7110.65) 4-8-1 Go-Around / Missed Approach

"A pilot may execute a go-around at their discretion. ATC may also instruct a pilot to go around when necessary"
“관제사는 필요 시 복행을 지시할 수 있으며, 조종사는 안전상 필요 시 언제든지 복행을 수행할 수 있음”

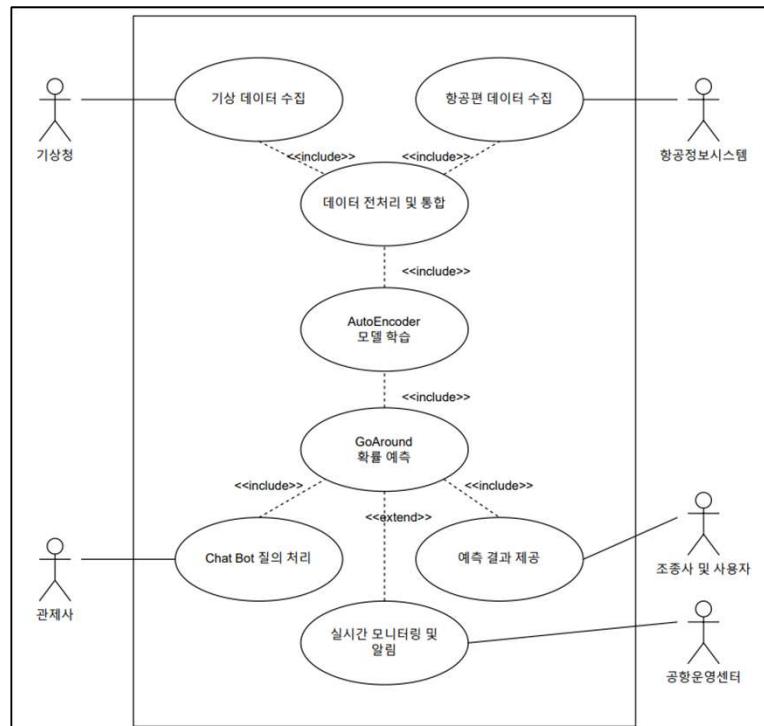
서비스에 대한 기획



AI Chatbot 사용 예시

- Generative AI Chatbot을 통해 조종사 및 관제사에게 실시간 예측과 대응 가이드를 제공하고자 하며, 조종사 및 관제사 뿐 아니라 일반 사람들도 쉽게 사용할 수 있도록 제작하고자 함
- 누구나 쉽게 접근할 수 있도록 웹 형식으로 진행할 것이며, API를 통해 기상 데이터를 받아 출력하고 질문자에게 해당 데이터를 가지고 지정하는 날짜의 착륙 복행(Go-Around)의 확률을 계산하여 전달해 주는 서비스

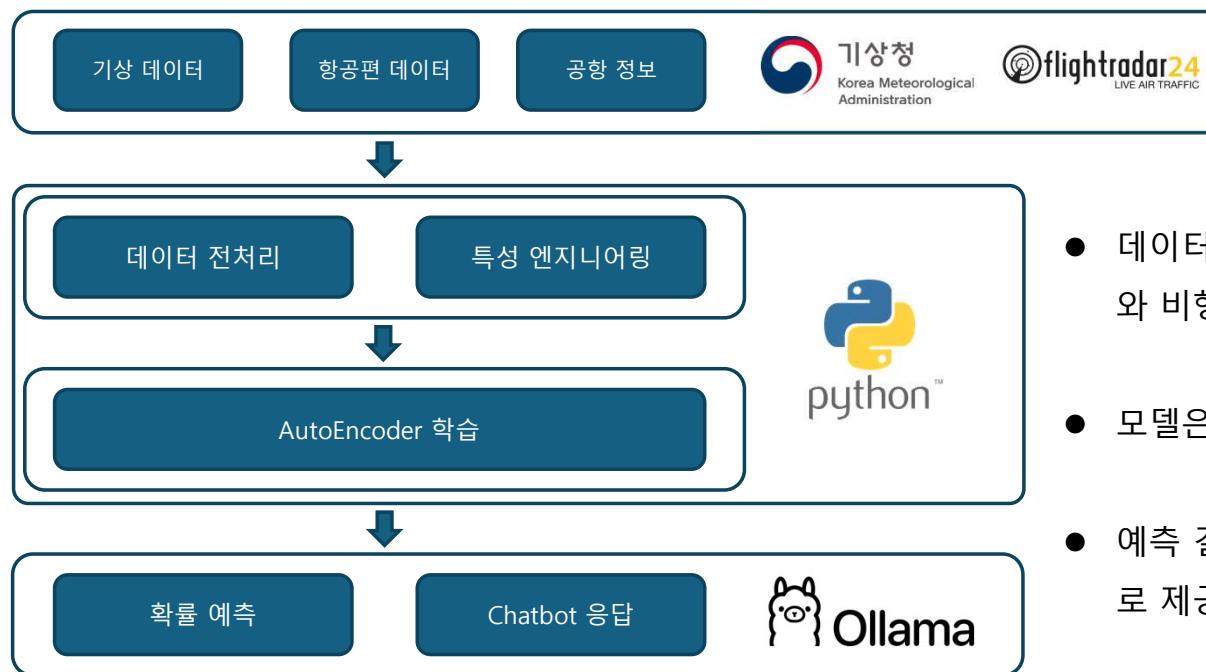
서비스에 대한 기획



서비스 유스케이스 다이어그램

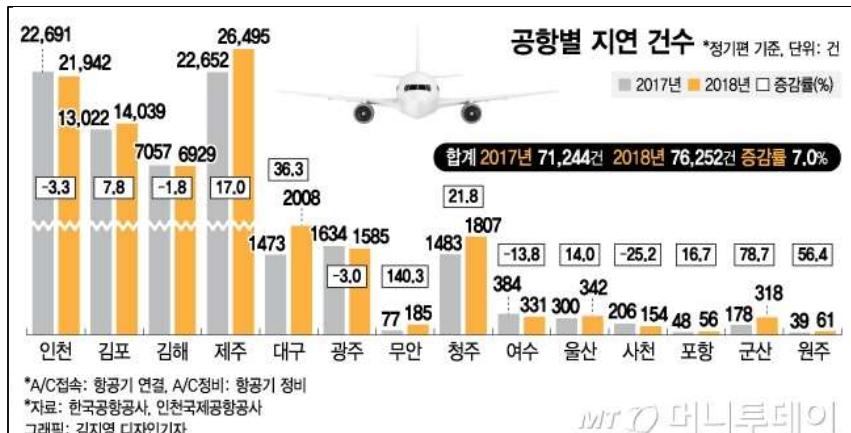
- 본 시스템은 기상 데이터와 항공편 운항 데이터를 통합하여, AutoEncoder 기반 모델을 통해 Go-Around(착륙 복행) 확률을 예측하고, 이를 Chatbot 기반 인터페이스를 통해 조종사·관제사·일반 이용객에게 제공
- 실시간 모니터링과 자연어 기반 질의응답을 통해 보다 빠르고 직관적인 항공 안전 정보 접근을 가능하게 함

서비스에 대한 기획



- 데이터는 기상청 및 flightradar24를 활용하여 기상 데이터와 비행 이력 및 운항 조건을 수집
- 모델은 Python으로 제작을 진행하여 학습 진행
- 예측 결과는 Chatbot 인터페이스를 통해 질의응답 형식으로 제공되며 Ollama를 사용하여 답변 제공

기획의도



<https://news.mt.co.kr/mtview.php?no=2019021016362729830>
미니투데이, "2018년 항공기 결항 ... ", 2019.02.11 문성일 선임기자

- Go-Around(착륙 복행)는 항공기 착륙 과정에서 발생하는 비정상 운항 상황으로, 조종사에게 큰 심리적 부담을 주며 항공편 지연이나 회항 등 운영상 혼란 유발
- 또한, Go-Around로 인해 발생하는 지연 여부는 일반 이용객이 사전에 파악하기 어려움
- 조종사의 의사결정 부담을 덜고, 탑승객이 사전에 항공편 지연 가능성을 확인할 수 있는 정보 제공형 Chatbot 시스템 구축

⇒ 이를 통해 **조종사와 관제사**의 판단 보조는 물론, 항공편을 이용하는 **일반 고객**이 비행기 운항 상황에 대해 사전 정보를 얻고 대비할 수 있는 경험을 제공하고자 함

2025 SW중심대학 디지털 경진대회

SW 부문

최종 산출물

팀명	DMS
대학명	창원대학교
서비스명(프로젝트명)	Captain AI

팀명	DMS
대학명	창원대학교
팀원(학과)	박진성(컴퓨터공학), 김성연(컴퓨터공학), 김남규(컴퓨터공학), 윤은옥(컴퓨터공학), 김민재(컴퓨터공학)
서비스명 (프로젝트명)	Captain AI
한줄소개 (서비스요약)	기상 데이터 기반 Go-Around 예측 모델을 통해 항공 안전 위험을 줄이기 위한 서비스
활용 AI	Ollama
주요 활용 데이터 (제공기관명)	기상 관련 정보(한국 기상청API) 항공 데이터(flighttrader24)

개발동기
및
서비스
제안 배경

- ✓ 기상 데이터 기반 Go-Around 예측 모델의 부재로 인한 항공 안전 위험을 줄이기 위해 서비스 기획
- ✓ Generative AI Chatbot을 통해 조종사 및 관제사에게 실시간 예측과 대응 가이드 제공

기대효과

- ✓ Go-Around 가능성 사전 예측으로 항공사고 위험 감소
- ✓ Generative AI Chatbot을 통한 실시간 안내로 조종사와 관제사의 대응 효율성 향상

개발동기

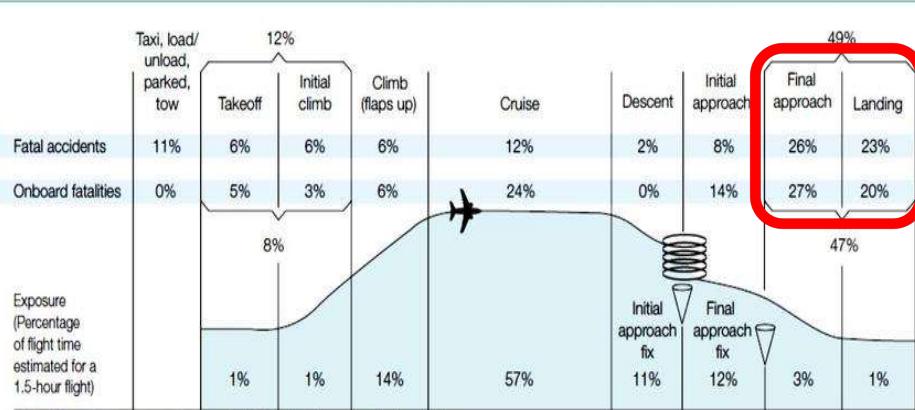
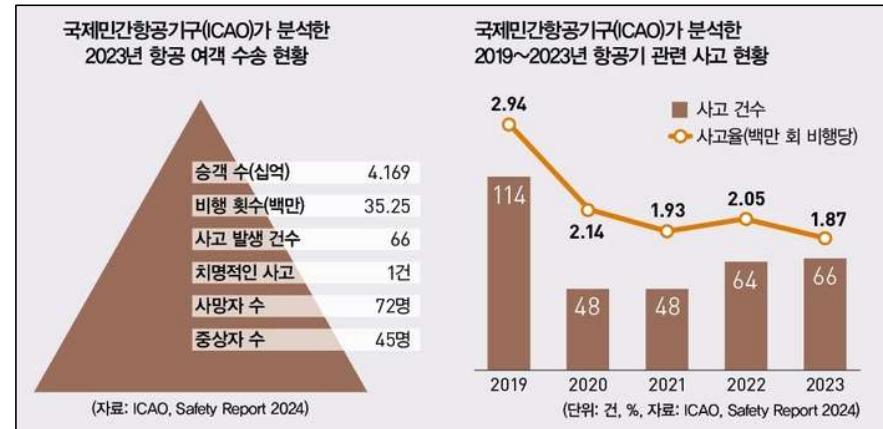


FIGURE 1 Percentage of fatal accidents and onboard fatalities (Boeing, 2015).

Donavalli, Bharath, "IMPACT OF WEATHER FACTORS ON GO-AROUND OCCURRENCE" (2016). Civil Engineering Theses. 445.

항공 단계별 사고 발생률



https://weekly.khan.co.kr/khnm.html?mode=view&art_id=202501060600031&code=115
주간경향, 원인 규명에 시간 필요.."조용히 기다려야", 2025.01.13, 김찬호 기자

항공 여객 수송 현황

착륙 단계에서 전체 항공사고의 49%가 발생하며 과거보다 항공기 사고 빈도는 분명히 줄어들고 있으나, 최근 '제주항공 여객기 참사'와 같은 사고가 발생한다면 탑승객의 생존 확률은 높지 않음 ⇒ 사고 빈도를 최대한 줄일 수 있도록 도와주는 AI 프로젝트 계획

개발동기



Federal Aviation
Administration

- 착륙 복행은 안전한 착륙이 어렵다고 조종사 또는 관제사가 주관적 판단으로 즉시 수행되는 절차
- 현재 조종사 및 관제사의 경험과 현장 상황에 의존하여 시행되고 있어, 어려움 존재
- 이를 돋기 위한 서비스를 개발함으로써 조종사의 부담을 줄이고자 함

① ICAO Annex 6 (Operation of Aircraft) 4-6-4 Approach and Landing

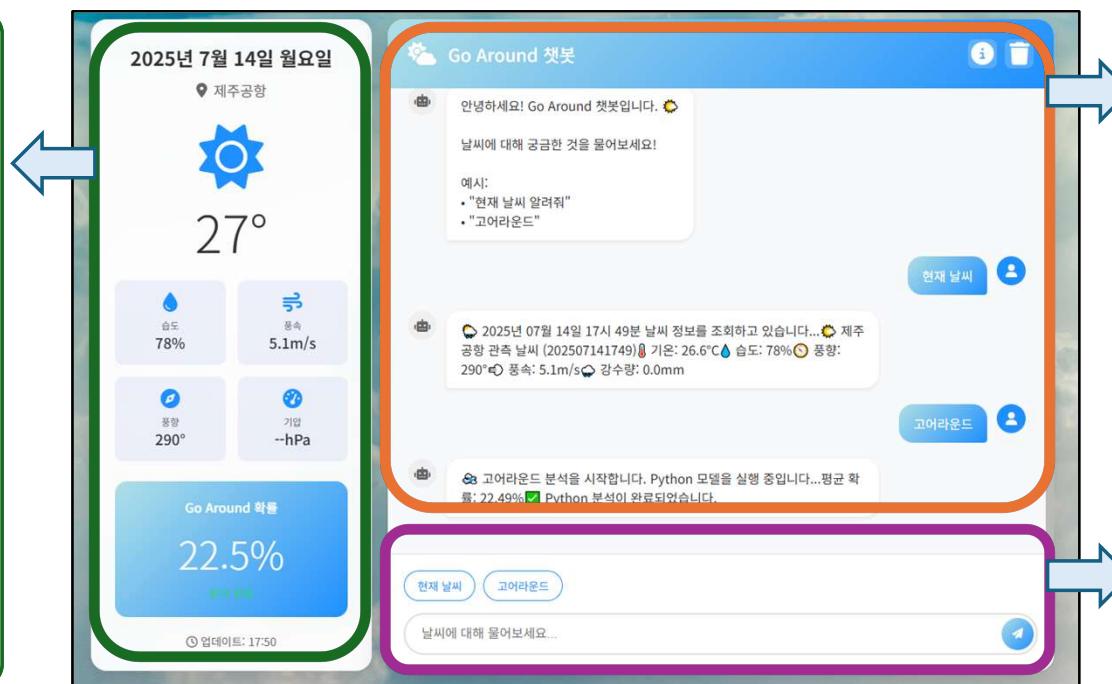
"If an approach cannot be continued safely, a missed approach shall be executed"
“안전하게 접근을 계속할 수 없다고 판단 시 복행을 조종사가 즉시 수행해야 함”

② FAA ATC Handbook (FAA JO 7110.65) 4-8-1 Go-Around / Missed Approach

"A pilot may execute a go-around at their discretion. ATC may also instruct a pilot to go around when necessary"
“관제사는 필요 시 복행을 지시할 수 있으며, 조종사는 안전상 필요 시 언제든지 복행을 수행할 수 있음”

서비스 개발 결과

- 입력한 날짜의 날씨 및 기상 상태를 업데이트
- 기상 상태에 따라 아이콘 변경
- 사용자의 입력에 따라 Go-Around 확률 분석 후 업데이트
- 업데이트 시간은 아래에 표기



- Ollama를 통한 사용자 질의 처리 및 응답화면 업데이트
- 기상 날씨 요청에 따른 결과 출력 및 분석 결과를 업데이트

- 사용자가 질의를 할 수 있는 인터페이스
- 편의를 위한 버튼형 대화 기능

개발 서비스 구현 과정

1. 서비스 구상 및 개발 배경

- 조종사 및 관제사는 착륙 직전 기상 변화에 의한 **Go-Around(착륙 복행)** 발생 가능성을 실시간으로 예측하기 어려움
- 기존 시스템에는 **기상 기반의 예측 모델이 부재**
- **데이터 기반 예측 시스템 + 자연어 대응 가이드를 손쉽게 제공할 수 있는 수단 필요** ⇒ 웹 구현

구성 요소	사용 기술	선택 배경
웹 서버	Node.js + Express	빠른 비동기 API 처리 및 구조 유연성
지도학습	XGBoost	정확도 우수, 결측치/변수 중요도 지원
비지도학습	AutoEncoder	Go-Around 불균형 보완, 이상탐지
Generative AI	Ollama	예측 결과를 자연어로 변환하여 직관적으로 제공
UI	Chatbot (웹 기반)	실시간 질의응답, 사용 편의성 강화

개발 서비스 구현 과정

2. 데이터 활용



- Go-Around(착륙 복행)가 발생한 항공 데이터
- 복행이 발생한 지점의 날씨 데이터

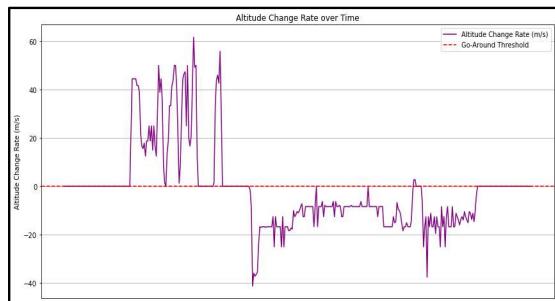
⇒ 이를 통한 예측 변수 및 학습 레이블 생성

- 착륙 복행이 발생하는 요인으로 지역적 특징이 가장 많이 영향을 끼치기 때문에 특정 공항을 기준으로 학습 진행 필요
- 전체 운항 예정 건수 대비 자연 건수의 비율이 가장 높은 제주 공항을 대상으로 진행
- 제주공항의 자연율은 24.0%로 김포공항(25.1%)과 인천공항(24.8%) 다음으로 높은 수준



개발 서비스 구현 과정

2. 데이터 활용



- 출발지 라인(의사결정고도)에 비행기가 닿는 시점의 날씨 데이터 전체를 학습
- 학습 후 위도 경도로 뽑아낸 Go-Around 시간의 날씨 데이터로 테스트 진행

계기 착륙 방식 CAT 1에 따르면 *의사결정고도는 200ft(60m)

*의사결정고도: 기장이 Go-Around를 결정하는 고도

개발 서비스 구현 과정

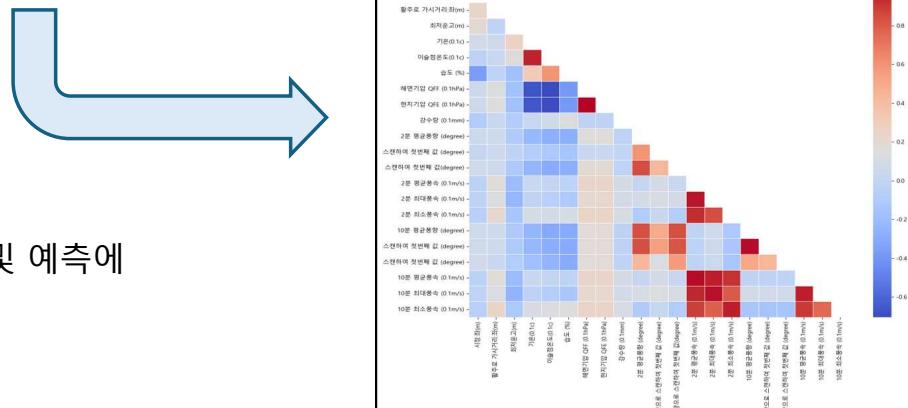
2. 데이터 활용

지점 코드	관측시 간(LST)	사정과 (m)	발로과 거리(m)				기온 (0.1°C)	이슬점 온도 (0.1°C)	습도 (%)	예비기압 OFF (0.1hPa)	현지기압 QFE (0.1hPa)	2분 평균기압 을 기준으로 시사면밀 정으로 소수점 첫번째 자(°degrees)				2분 평균풍 속 (0.1m/s)	2분 최대풍 속 (0.1m/s)	2분 최소풍 속 (0.1m/s)	10분 평균풍 향 각(degree)	10분 평균향 향 각(degree)	10분 평균향 향 각(degree)	10분 평균향 향 각(degree)
			최저온 도(m)	기온 도(m)	이슬점 온도 도(m)	습도 도(m)						2분 평균기압 을 기준으로 시사면밀 정으로 소수점 첫번째 자(°degrees)										
0	182 0-01-0000	10000	2000	3300	84	6	58	10262	10231	...	350	38	55	19	340	360	340	37	63	16	16	
	2024- 01-01-0001	10000	2000	3200	84	7	59	10262	10231	...	350	27	41	16	340	360	340	34	63	16	16	
1	182 0-01-0001	10000	2000	3200	84	8	59	10262	10231	...	350	31	52	16	340	360	340	34	63	16	16	
	2024- 01-01-0002	10000	2000	3200	84	8	59	10262	10231	...	340	42	71	25	340	360	340	36	71	16	16	
3	182 0-01-0003	10000	2000	3200	83	8	58	10262	10230	...	340	42	71	25	340	360	340	36	71	16	16	
	2024- 01-01-0004	10000	2000	3200	83	6	57	10262	10231	...	310	45	71	25	340	360	340	37	71	16	16	

⇒ 상관관계 히트맵을 통해 25개에서 14개 컬럼 추출

모델 성능 향상을 위해 기상 데이터 전체 25개의 컬럼 중 분석 및 예측에
유의미한 변수만 추출

- flightrader24 API로 제주공항의 1년치 항공 데이터를 시간별로 수집
 - 총 5904개의 데이터 중 실제 활주로에 진입하지 않은 35개 데이터는 제외



개발 서비스 구현 과정

2. 데이터 활용

TABLE 1 Proportion of Go-Arounds of Each Individual Variable Range				
Variable	Range	Total Arrivals	Go-arounds	Proportion of Go-around
Wind Gust	High	4810	25	0.005198
	Medium	5399	10	0.001852
	Low	5600	12	0.002143
Event	Thunderstorm	4974	21	0.004222
	Rain-Snow	5189	11	0.002120
	No event	5646	15	0.002657
Visibility	High visibility	8032	23	0.002864
	Low visibility	7777	24	0.003086

Donavalli, Bharath, "IMPACT OF WEATHER FACTORS ON GO-AROUND OCCURRENCE" (2016). Civil Engineering Theses. 445.



TABLE 2 Two Proportion Z Tests				
Variable	Comparing (Range)	Hypothesis	Result P-value	Remark
Wind Gust	High	Null: $p_1 - p_2 = 0$	0.0025	Go-around rate during high wind gust is significantly greater than medium range
	Medium	Alternative: $p_1 - p_2 > 0$		
Wind Gust	Medium	Null: $p_1 - p_2 = 0$	0.7327	No significant difference between medium and low range wind gust
	Low	Alternative: $p_1 - p_2 \neq 0$		
Wind Gust	High	Null: $p_1 - p_2 = 0$	0.0057	Go-around rate during high wind gust is significantly greater than low range
	Low	Alternative: $p_1 - p_2 > 0$		
Event	Thunderstorm	Null: $p_1 - p_2 = 0$	0.0302	Go-around rate during thunderstorm event is significantly greater than the rate during rain-snow events
	Rain-Snow	Alternative: $p_1 - p_2 > 0$		
Event	Rain-Snow	Null: $p_1 - p_2 = 0$	0.5664	No significant difference between rain-snow events and no events
	No event	Alternative: $p_1 - p_2 \neq 0$		
Event	Thunderstorm	Null: $p_1 - p_2 = 0$	0.0861	Go-around rate during thunderstorm event is significantly greater than the rate during no events
	No event	Alternative: $p_1 - p_2 > 0$		
Visibility	High	Null: $p_1 - p_2 = 0$	0.7974	No significant difference between high and low visibility
	Low	Alternative: $p_1 - p_2 \neq 0$		



강수량 Z-test 결과	
[강수량] High vs Low:	Z = 3.0107, p = 0.0026063958
[강수량] High vs Medium:	Z = 1.3171, p = 0.1878114195
[강수량] Medium vs Low:	Z = 0.2131, p = 0.8312555141
기온 Z-test 결과	
[기온] High vs Low:	Z = 2.8738, p = 0.0040555436
[기온] High vs Medium:	Z = 2.4384, p = 0.0147542470
[기온] Medium vs Low:	Z = 1.1974, p = 0.2311600735
이슬점온도 Z-test 결과	
[이슬점온도] High vs Low:	Z = 2.9331, p = 0.0033561677
[이슬점온도] High vs Medium:	Z = 0.7758, p = 0.4378537780
[이슬점온도] Medium vs Low:	Z = 1.8068, p = 0.0707996986
해면기압 Z-test 결과	
[해면기압] High vs Low:	Z = -4.1024, p = 0.0000408814
[해면기압] High vs Medium:	Z = -1.1874, p = 0.2350668577
[해면기압] Medium vs Low:	Z = -3.1619, p = 0.0015673386
현지기압 Z-test 결과	
[현지기압] High vs Low:	Z = -3.6852, p = 0.0002285163
[현지기압] High vs Medium:	Z = -2.2583, p = 0.0239238679
[현지기압] Medium vs Low:	Z = -1.8161, p = 0.0693531811
2분 평균풍속 Z-test 결과	
[2분 평균풍속] High vs Low:	Z = 6.9496, p = 0.0000000000
[2분 평균풍속] High vs Medium:	Z = 2.8806, p = 0.0039685736
[2분 평균풍속] Medium vs Low:	Z = 4.2238, p = 0.0000240259

- 위의 논문을 참고하여 14개 컬럼들의 범위를 나누어 총 도착횟수 대비 Go-Around 확률 계산
- 범주 14개를 통해 Range 별로 Z-Test 진행

p-value가 작고 |Z|가 큰 비교 = 두 그룹 간 고어라운드 비율 차이가 통계적으로 유의미

개발 서비스 구현 과정

2. 데이터 활용

강수량 Z-test 결과
[강수량] High vs Low: Z = 3.0107, p = 0.0026063958
[강수량] High vs Medium: Z = 1.3171, p = 0.1878114195
[강수량] Medium vs Low: Z = 0.2131, p = 0.8312555141

기온 Z-test 결과
[기온] High vs Low: Z = 2.8738, p = 0.0040555436
[기온] High vs Medium: Z = 2.4384, p = 0.0147542470
[기온] Medium vs Low: Z = 1.1974, p = 0.2311600735

이슬점온도 Z-test 결과
[이슬점온도] High vs Low: Z = 2.9331, p = 0.0033561677
[이슬점온도] High vs Medium: Z = 0.7758, p = 0.4378537780
[이슬점온도] Medium vs Low: Z = 1.8068, p = 0.0707996986

해면기압 Z-test 결과
[해면기압] High vs Low: Z = -4.1024, p = 0.0000408814
[해면기압] High vs Medium: Z = -1.1874, p = 0.2350668577
[해면기압] Medium vs Low: Z = -3.1619, p = 0.0015673386

현지기압 Z-test 결과
[현지기압] High vs Low: Z = -3.6852, p = 0.0002285163
[현지기압] High vs Medium: Z = -2.2583, p = 0.0239238679
[현지기압] Medium vs Low: Z = -1.8161, p = 0.0693531811

2분 평균풍속 Z-test 결과
[2분 평균풍속] High vs Low: Z = 6.9496, p = 0.0000000000
[2분 평균풍속] High vs Medium: Z = 2.8806, p = 0.0039685736
[2분 평균풍속] Medium vs Low: Z = 4.2238, p = 0.0000240259



2분 최소풍속 Z-test 결과
[2분 최소풍속] High vs Low: Z = -0.2446, p = 0.8068033527
[2분 최소풍속] High vs Medium: Z = -0.4283, p = 0.6684165346
[2분 최소풍속] Medium vs Low: Z = 3.8122, p = 0.0001377223

2분 최소풍속은 High 조건에서 Go-Around 비율과 큰 관계가 없음

Medium을 Low와 비교했을 때 유의미한 차이가 있었음

즉, 최소풍속보다 평균풍속, 최대풍속이 더 중요한 요인임을 시사

"최저운고(m)",
"기온(0.1c)",
"이슬점온도(0.1c)",
"해면기압 QFF (0.1hPa)",
"현지기압 QFE (0.1hPa)",
"강수량 (0.1mm)",
"2분 평균풍속 (0.1m/s)",
"2분 최대풍속 (0.1m/s)",
"2분 최소풍속 (0.1m/s)",
"10분 평균풍향 (degree)",
"10분 최우풍향 (degree)",
"10분 평균풍속 (0.1m/s)",
"10분 최대풍속 (0.1m/s)",
"10분 최소풍속 (0.1m/s)"



"최저운고(m)",
"기온(0.1c)",
"이슬점온도(0.1c)",
"해면기압 QFF (0.1hPa)",
"현지기압 QFE (0.1hPa)",
"강수량 (0.1mm)",
"2분 평균풍속 (0.1m/s)",
"2분 최대풍속 (0.1m/s)",
"10분 평균풍향 (degree)",
"10분 최우풍향 (degree)",
"10분 평균풍속 (0.1m/s)",
"10분 최대풍속 (0.1m/s)"

기준 14개 컬럼에서 최소풍속 컬럼을 제거하여 12개 컬럼으로 축소

개발 서비스 구현 과정

3. 예측 모델

- 모델은 지도 학습, 비지도 학습으로 총 두 가지를 사용
- 두 모델의 결과 값을 토대로 Go-Around 확률 계산

지도학습

- 정답이 있는 데이터를 활용해 데이터를 학습하는 방식

회차	기준	이슬람교도	예수교인(기독교)	무슬림(이슬람교)	기준						
(n)	(0.1s)	(0.5s)	(0.5s)	(0.5s)	(0.1ms)						
0	7.803377	4.369448	2.772589	9.239962	9.239959	0.000000	3.970392	4.395773	3.419887	3.433087	3.784790
1	7.901377	4.369448	2.772589	9.239962	9.239959	0.000000	3.970392	4.395773	3.419887	3.433087	3.409710
2	8.038439	4.341895	1.695438	9.237777	9.234852	0.000000	2.899172	3.131205	3.911826	3.181826	3.258997
3	8.077219	4.340665	1.791759	9.237074	9.234252	0.000000	2.468907	3.135444	5.017280	5.081644	2.484407
4	8.165084	4.264469	5.940710	9.237664	9.236443	0.000000	3.091042	3.252967	5.017280	5.081644	3.178954
5	8.204031	4.126631	1.196497	4.852030	9.236106	9.233178	0.000000	4.394449	4.664349	4.705460	4.841542
6	8.242756	4.031599	3.806562	9.232982	9.229947	0.000000	3.920452	4.045570	4.459250	4.429549	4.429549
5805	8.242756	4.031599	3.806562	9.232982	9.229947	0.000000	3.461736	3.931804	5.768321	5.880158	3.355348
5806	8.268732	4.295459	2.772589	9.234838	9.237170	3.091942	4.330733	4.779312	5.703782	5.828941	3.404965
5807	7.974666	4.515960	4.204935	9.235083	9.227951	0.000000	4.295469	4.915265	5.786572	5.857931	4.366448

```
class ImprovedAutoencoder(nn.Module):
    def __init__(self, input_dim, encoding_dim=4):
        super(ImprovedAutoencoder, self).__init__()
        self.input_dim = input_dim
        self.encoding_dim = encoding_dim

        # Encoder: 12 → 8 → 4
        self.encoder = nn.Sequential(
            nn.Linear(input_dim, 8),
            nn.BatchNorm1d(8),
            nn.ReLU(),
            nn.Dropout(0.1),

            nn.Linear(8, encoding_dim),
            nn.BatchNorm1d(encoding_dim),
            nn.ReLU()
        )

        # Decoder: 4 → 8 → 12
        self.decoder = nn.Sequential(
            nn.Linear(encoding_dim, 8),
            nn.BatchNorm1d(8),
            nn.ReLU(),
            nn.Dropout(0.1),
            nn.Linear(8, input_dim)
        )
```

AutoEncoder 코드

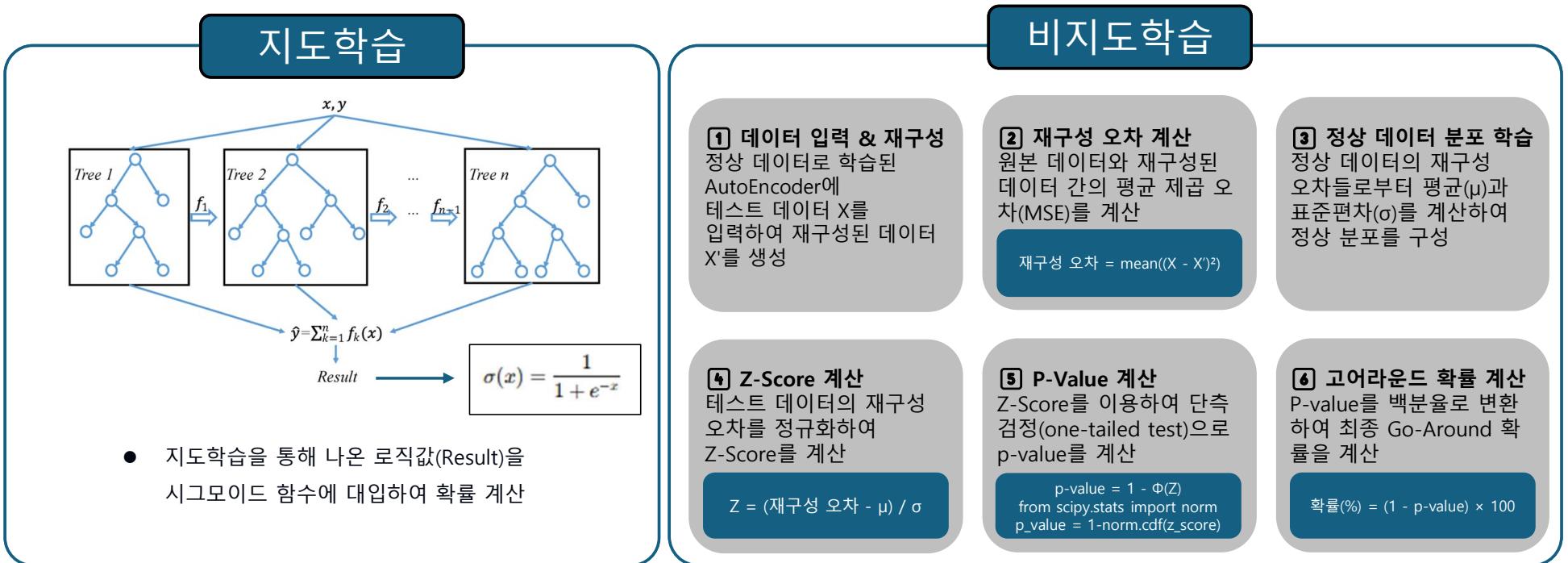
비지도학습

- 정답 라벨이 없는 데이터를 비슷한 특징끼리 군집하여 새로운 데이터에 대한 결과를 예측하는 방식



개발 서비스 구현 과정

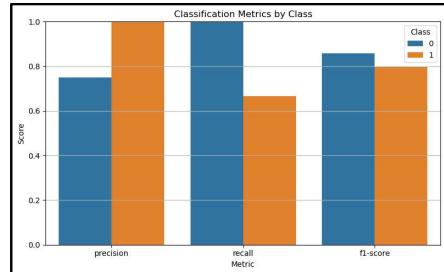
3. 예측 모델(Go-around 발생 확률 도출 과정)



개발 서비스 구현 과정

3. 예측 모델

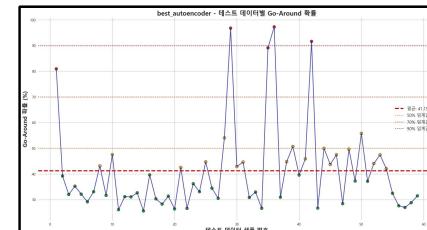
지도학습



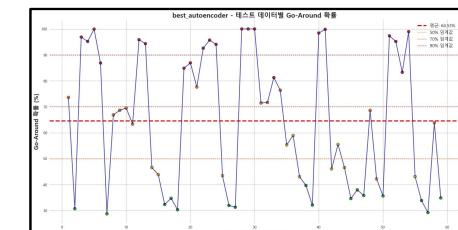
지도학습 모델 지표

- 테스트의 경우 클래스 0인 경우 18가지, 클래스 1인 경우 18가지(전체의 20%)로 1:1 비율의 데이터로 진행
- Precision은 클래스 1(Go-Around), 클래스 0(Not Go-Around) 각각 1.0과 0.75의 결과를 가짐
- Recall의 경우 각각 1.0, 0.67의 결과를 가졌으며, F1점수의 경우 클래스 0은 0.86 클래스 1의 경우 0.80으로 전체 정확도(Accuracy)는 0.83으로 비교적 양호한 성능을 나타냄

비지도학습



Not Go-Around 59개



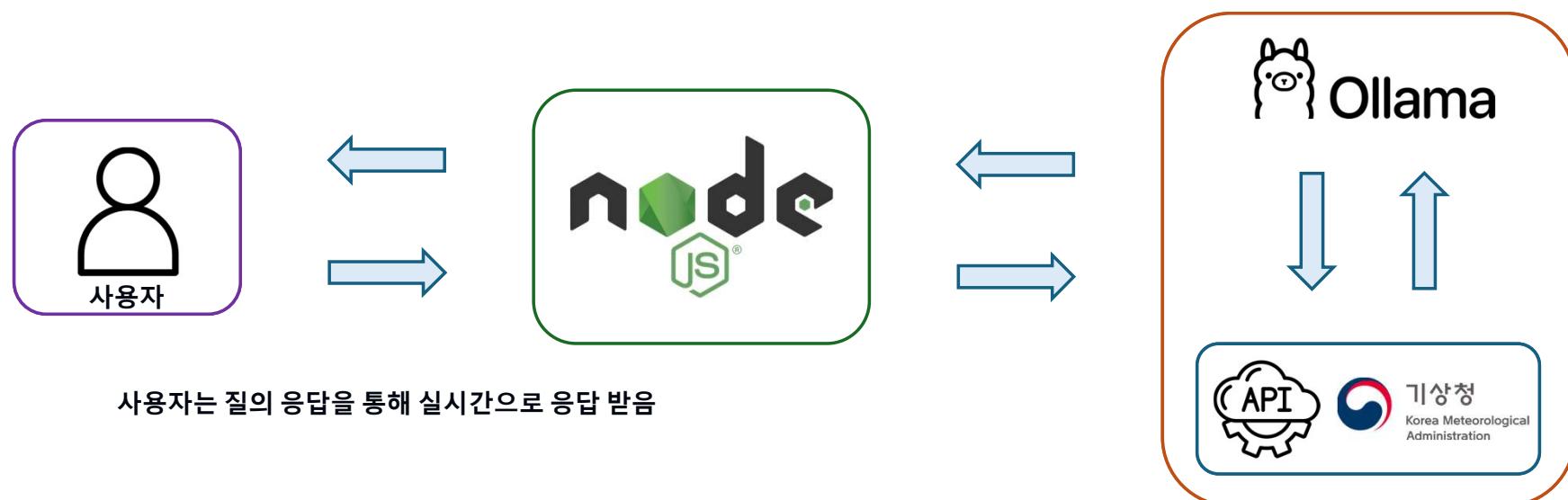
Go-Around 59개

- 각 점은 개별 샘플의 예측된 Go-Around 확률값을 나타냄
- Go-Around가 발생한 59개 중 30개로 50%의 정확도를 가졌으며, 발생하지 않은 경우 59개 중 23개로 62%의 정확도를 나타냄

개발 서비스 구현 과정

4. 웹 개발 (요구사항)

사용자의 질문에 대한 자연어 처리가 가능하며, 이에 대응할 수 있는 AI가 필요



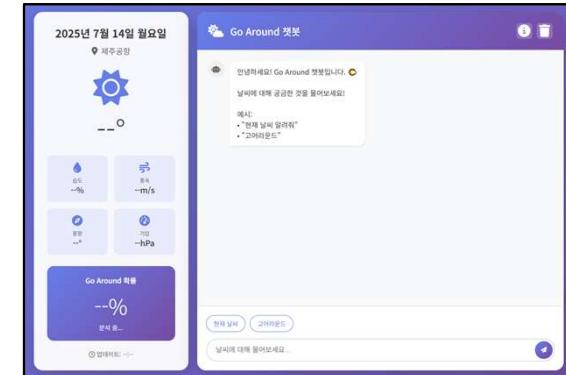
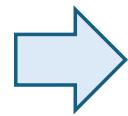
개발 서비스 구현 과정

4. 웹 개발

사용자는 질의 응답을 통해 실시간으로 응답 받음

사용자의 질문에 대한 자연어 처리가 가능하며,
이에 대응할 수 있는 AI가 필요

사용자의 질문에 맞는 데이터를 가져와 사용



- 해당 요구 조건을 만족하기 위해 Node.js 기반 SSE서버를 구축
- UI는 CSS를 통해 구성
- /stream 엔드 포인트를 통해 서버는 사용자의 질문 처리를 스트리밍 방식으로 지속 전달

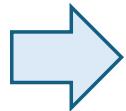
개발 서비스 구현 과정

4. 웹 개발

사용자는 질의 응답을 통해 실시간으로 응답 받음

사용자의 질문에 대한 자연어 처리가 가능하며,
이에 대응할 수 있는 AI가 필요

사용자의 질문에 맞는 데이터를 가져와 사용



The screenshot shows the Ollama web application. On the left, there is a code editor window displaying the following JavaScript code:

```
const OLLAMA_HOST = process.env.OLLAMA_HOST || 'localhost';
const OLLAMA_PORT = process.env.OLLAMA_PORT || 11434;
const OLLAMA_MODEL = process.env.OLLAMA_MODEL || 'mistral';
```

On the right, there is a conversational interface. The top part shows a message from the user: "안녕하세요" (Hello) and a response: "반갑습니다" (Nice to meet you). Below that, it shows the current time: "현재는 12:00입니다. (It's currently 12 PM.)". The bottom part is labeled "질의 응답 처리" (Question Response Processing) and contains a bulleted list of features.

- Ollama는 로컬에서도 대규모 언어 모델을 실행할 수 있도록 도와주는 오픈소스 플랫폼
- 해당 모델을 통해 사용자의 자연어 처리 담당
- 인터넷 연결없이 자유로우며, 사용량에 따른 추가비용이 발생하지 않음
- 다양한 모델들 중 'mistral'을 사용

개발 서비스 구현 과정

4. 웹 개발

사용자는 질의 응답을 통해 실시간으로 응답 받음

사용자의 질문에 대한 자연어 처리가 가능하며,
이에 대응할 수 있는 AI가 필요

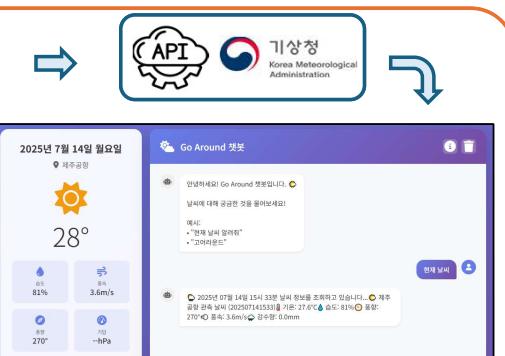
사용자의 질문에 맞는 데이터를 가져와 사용

```
기준 세션 사용: 02bbf739-999c-4d98-925d-2cbde67682c7
채팅 요청 - 세션: 02bbf739-999c-4d98-925d-2cbde67682c7, 메시지: 현재 날씨
날씨 요청 처리 - 세션: 02bbf739-999c-4d98-925d-2cbde67682c7
날씨 조회 요청: {
  sessionId: '02bbf739-999c-4d98-925d-2cbde67682c7',
  userMessage: '현재 날씨',
  dateStr: '20250714',
  targetTime: { hour: '15', minute: '18' },
  dateDisplay: '2025년 07월 14일 15시 18분',
  isSimpleWeatherRequest: true
}

async function fetchKmaWeather(tm, dtw1, stn = 182) {
  const url = `https://api.kma.go.kr/api/tms/v1/imos.php?tm=202211301200&dtw=${dtw1}&stn=${stn}&key=pbLwQ6jHmzg8Eoz25rnew`;
  try {
    const res = await axios.get(url, {
      params: {
        authKey: process.env.KMA_API_KEY,
        tm,
        dtw,
        stn,
        help: '1',
      },
      responseType: 'text',
    });
  
```

질의에 따른 API 요청

- Ollama는 로컬에서도 대규모 언어 모델을 실행할 수 있도록 도와주는 오픈소스 플랫폼
- 해당 모델을 통해 사용자의 자연어 처리 담당
- 인터넷 연결없이 자유로우며, 사용량에 따른 추가비용이 발생하지 않음
- 다양한 모델들 중 'mistral'을 사용



API 결과물 출력

개발 서비스 장점

- 기존 유사 서비스와의 차별성

항목	기존 날씨 서비스 (ex/ 기상청 앱, 포털 날씨)	개발한 서비스 (Captain AI)
정보 접근 방식	메뉴 탐색, 수치 중심 표시	자연어 질문 기반 실시간 대화형 응답
예측/분석 가능	단순 관측치 제공	자체 AI 모델 기반 분석 결과 제공
지역 특화성	전국 단위 중심	제주공항 특화 정보 제공
UX/UI	정적 웹/앱 UI	스트리밍 기반 대화형 인터페이스 (SSE)
확장성	제한적	RAG, LLM 연계로 범용 정보 질의 응답 가능

개발 서비스 주제 적합성

전국 공항 항공기 결항 통계 *단위: 건, ()안은 비율 %										
구분	운항	기상	A/C 접속	A/C 정비	여객 처리	복합 원인	기타	총계	전년대비 증감률	운항횟수 대비 결항률
2018년	86만 4,422	4,359 (64.8)	882 (13.1)	260 (3.9)	782 (11.6)	1 (0.0)	443 (6.6)	6,727 (100.0)	187.5%	0.78%
2017년	82만 7,535	1,240 (53.0)	532 (22.7)	213 (9.1)	123 (5.3)	- (0.0)	232 (9.9)	2,340 (100.0)	-66.0%	0.28%

*A/C접속: 항공기 연결, A/C정비: 항공기 정비
 *자료: 한국공항공사, 인천국제공항공사
 그래픽: 김지영 디자인기자

MTD 머니투데이

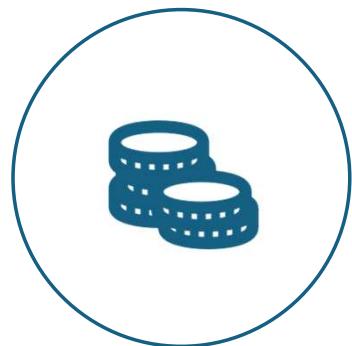
<https://news.mt.co.kr/mtview.php?no=2019021016362729830>
 머니투데이, 2018년 항공기 결항 전년보다 1.9배 급증..., 2019.02.11, 문성일 선임기자

- 본 서비스는 제주공항 지역의 실시간 날씨 데이터를 바탕으로, 사용자의 자연어 질의에 맞춰 Generative AI가 이해하고 응답하는 공공형 Chatbot 시스템
- 기후 변화와 이상 기후로 인한 항공편 지연, 결항, 안전 문제가 빈번히 발생하는 상황에서 기상 정보 격차로 어려움을 겪는 여행자, 지역주민, 조종사 등의 불안과 혼란을 줄이는 데 직접적으로 기여
- 단순한 Chatbot을 넘어, 실시간 공공 데이터 + 사용자 문맥 이해 + AI 예측 모델을 융합하여 기후 정보에 따른 위험 대응 및 정보 접근성 불균형을 해결하는데 기여하며, Generative AI 기술이 사회 문제 해결에 실질적으로 적용되는 대표적인 사례라 할 수 있음

개발 서비스의 기대효과



항공 안전성 강화



운항 효율성



데이터 기반 인프라 발전



연료 배출 절감

돌발 기상 상황에서 즉각적인 의사결정 가능
Go-Around 및 연쇄 지연 상황 감소

불필요한 공중 대기, Go-Around로 인한 연료 낭비 감소
⇒ 항공사 운영비 절감

축적된 운항·기상 데이터로 공항별 안전 개선 가능

불필요한 연료 소모 방지로 탄소 배출 감소

개발 서비스의 기대효과

- 상용화 계획



1단계

2단계

3단계

4단계

App 개발

모바일 기기 사용
자를 위한 맞춤 UI
반영한
Application 개발

안정화, 피드백

사용자의 피드백을
기반으로 개선 및
사용자 인터페이스
고도화

타 공항 확장

제주 공항 이외의
국내 타 공항에서
의 Go-Around 확
률도 예측

해외 공항 진출

아시아·유럽 등 해
외의 여러 공항 파
트너쉽 체결

개발 서비스의 편의성

- 자연어 질의 처리

사용자는 어려운 명령어나 버튼 없이 일상적인 문장으로 질문 가능

- 기상 데이터 연동

항공/여행/일상 활동 전 실시간 날씨 확인 가능

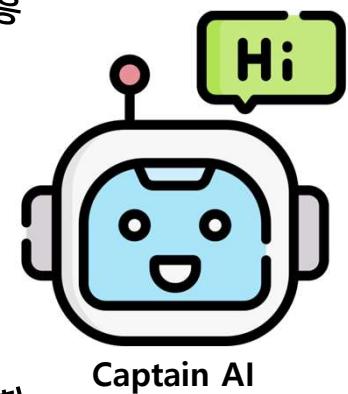
- 실시간 응답 스트리밍

사용자는 로딩 없이 단계별 응답을 실시간 수신

“오늘 바람 많이 불어요?” 등의 질문에 대응

복잡한 앱 없이 단일 창에서 즉시 확인

Chatbot과 대화하듯 즉각적인 피드백 경험



- **공항기상관측(AMOS)**
기상청 API허브(<https://apihub.kma.go.kr>)
- **항공 데이터**
flightrader24(<https://www.flightradar24.com/51.47,0.46/6>)
- **Approach and Landing**
ICAO Annex 6, Part I, Chapter 4, Section 6, Clause 4
- **Go-Around / Missed Approach**
FAA ATC Handbook Chapter 4, Section 8, Paragraph 1
- **항공 단계별 사고 발생률**
Donavalli, Bharath, "IMPACT OF WEATHER FACTORS ON GO-AROUND OCCURRENCE" (2016). Civil Engineering Theses. 445.
- **항공 여객 수송 현황**
주간경향, 원인 규명에 시간 필요.."조용히 기다려야", 2025.01.13, 김찬호 기자
(https://weekly.khan.co.kr/khnm.html?mode=view&art_id=202501060600031&code=115)

- **Node.js**
오픈 소스 자바스크립트 런타임 환경
Non-blocking I/O 이벤트 기반 아키텍처로 실시간 웹 애플리케이션 처리량과 확장성 최적화
- **Express.js**
Node.js 위에서 돌아가는 웹 애플리케이션 프레임워크
네트워킹, 요청 처리, 라우팅 등의 코딩을 간단하고 직관적으로 만듦
- **XGBoost**
지도학습에 사용하며 정답이 있는 데이터를 학습하여 새 데이터의 정답 예측
- **AutoEncoder**
비지도학습에 사용하며 입력 데이터를 압축(Encoder)하고 다시 복원(Decoder)하는 신경망
출력이 입력과 같도록 학습
- **Ollama**
대규모 언어 모델(LLM)을 Local에서 실행할 수 있는 도구
- **Mistral**
오픈소스 LLM
생성형 언어 모델로 텍스트 요약, 질문 답변 등 가능하며 성능 대비 모델 크기가 작고 빠름