S/W 시스템 설계서

Project Name	서울의 안전한 문화 생활을 위한 혼잡도 기반, 장소 추천 웹		
Date	2023-06-17	Version	1.0
변경 이력			
작성자	**전원 참여**	**승인자**	**Saemaro Moon**
조직명	StandUpSeoul		

제정및 개정 이력

버전	개정 내용	작성자	승인자	적용 날짜
1.0	최초 생성	우상욱	Saemaro Moon	2023.06.24
1.1	파트별 수정	전원 참여	Saemaro Moon	2023.06.25

목 차

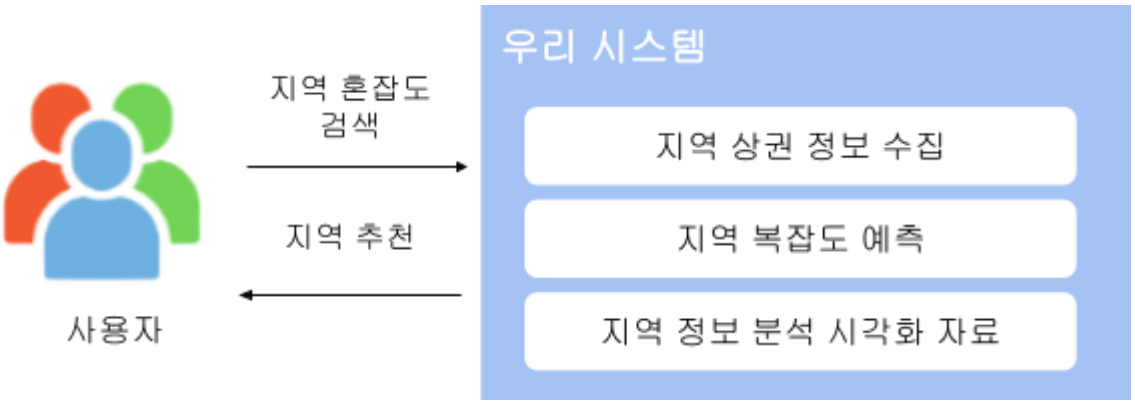
1. 개요

1.1. 목적

- 서울시 발생 사고로 인한 혼잡도에 대한 관심 급증
- 안전성과 여가·문화 생활을 결합한 종합 정보 제공 웹 서비스의 필요성
- 서울시 주요 장소 발생 혼잡 데이터 수집, 데이터 분석 및 ML 기반 예측 웹 서비스 제공

1.2. 시스템 개요

[시스템 개요 장표**]**



- 서울시에서 혼잡도가 낮은 주요 48곳의 장소에 대한 맛집, 명소, 문화 관광 정보 제공 및 추천
- 실시간으로 사용자가 선택한 장소를 기반으로 해당 장소와 가까운 여유 지역 추천
- 머신러닝 기반으로 한 혼잡도 분류 모델을 활용하여 특정 시간, 장소 예측 정보 제공
- 사용자의 맛집, 명소 리뷰 데이터를 활용하여 인근 지역의 맛집, 명소 추천 알고리즘 제공

1.3. 가정

- 지하철 하차 인원 비율을 역 마다의 사분위수를 파악한 후에 승하차인원에 따라 분류를 진행하여, 1사분위(Q1)를 여유, 2사분위(Q2)를 보통, 3사분위(Q3)를 약간 혼잡, 4사분위를 혼잡으로 분류하여 사분위수를 기준으로 혼잡도를 예측하여, 내가 가려고 하는 장소의 지하철역 혼잡하다면 장소가 혼잡하다고 가정한다.

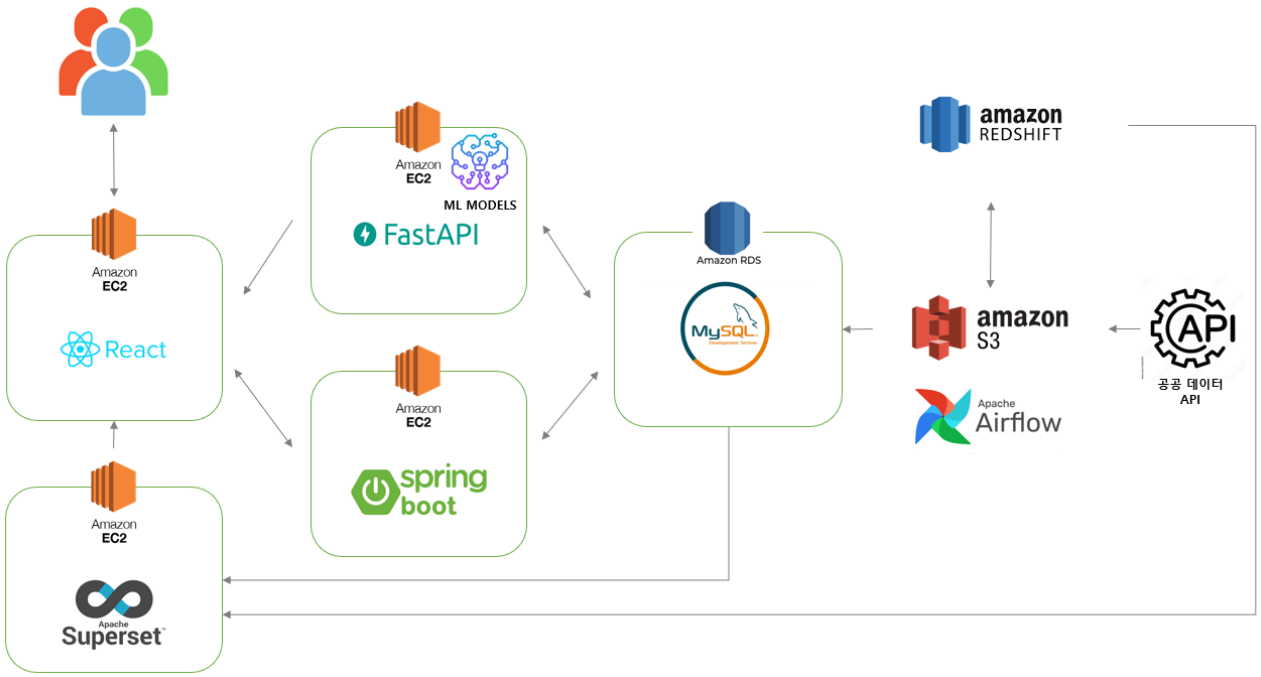
1.4. 제약사항

- 예측 feature가 범용적이어서 특별한 날(워터밤, 콘서트와 같은 특정 행사)에 대한 처리가 불가능함.
- 코로나에 따른 변화를 고려하여 2018~2023년도를 기준으로 모델 학습을 진행.
- 해당 장소의 혼잡한 정도는 승하차인원과 비례한다고 가정.

2. 시스템 구성

2.1. Overview

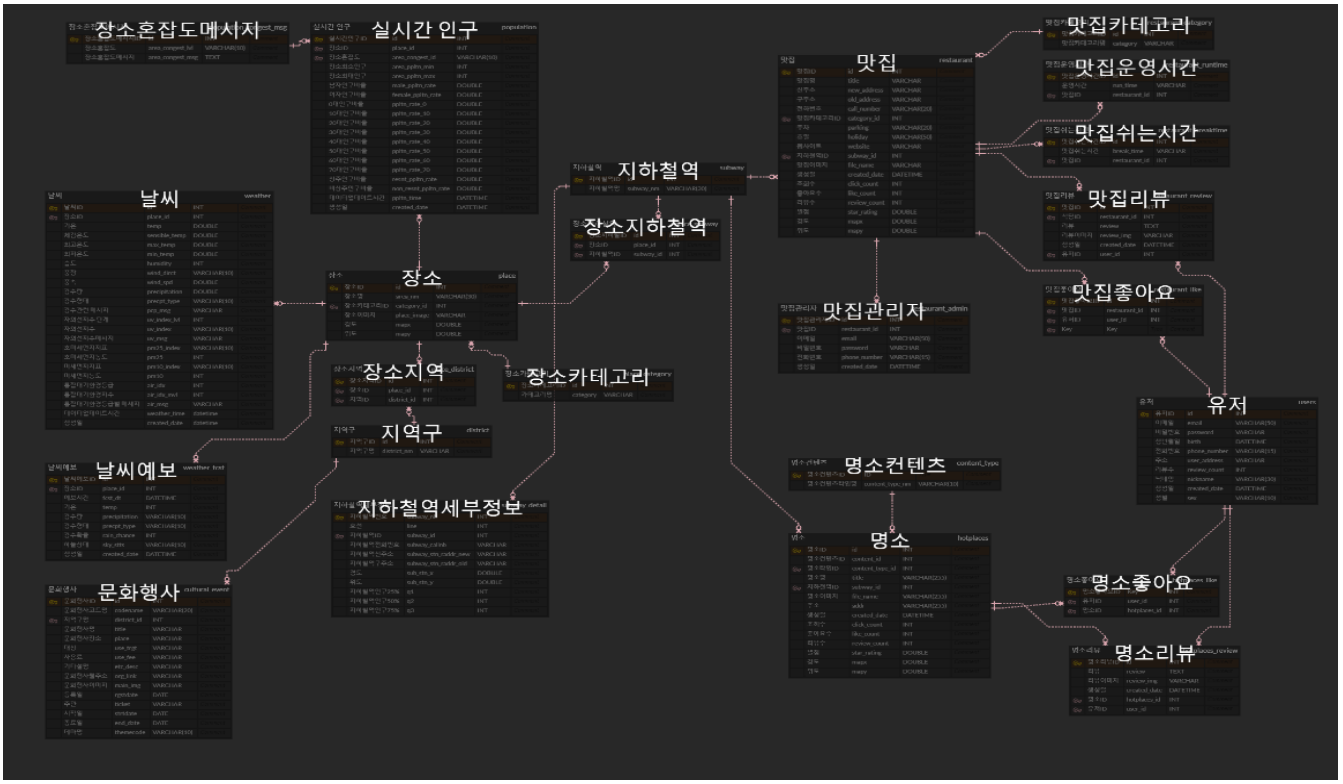
[전체 시스템 구성 장표**]**



2.2. ERD 설계

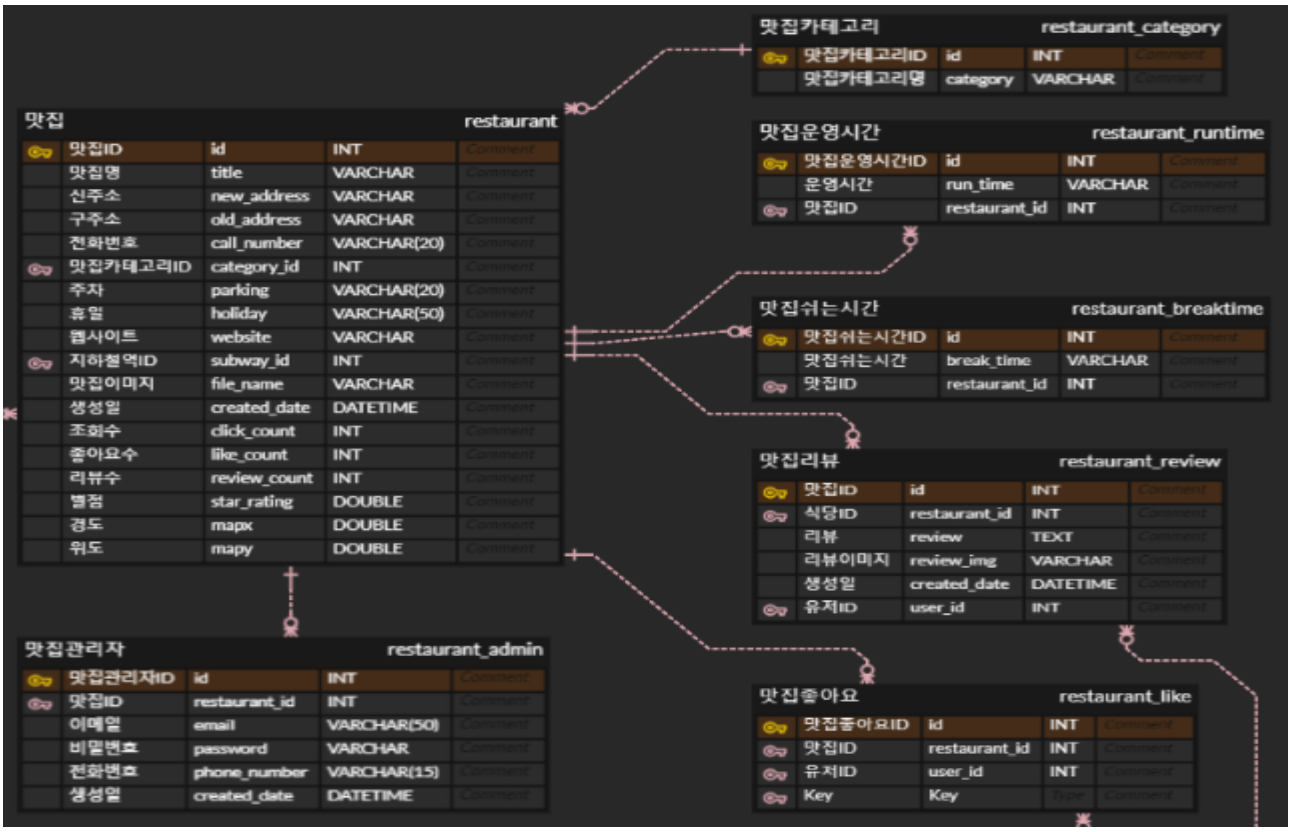
시스템 구성에 필요한 ERD 설계 내용

[전체**]**

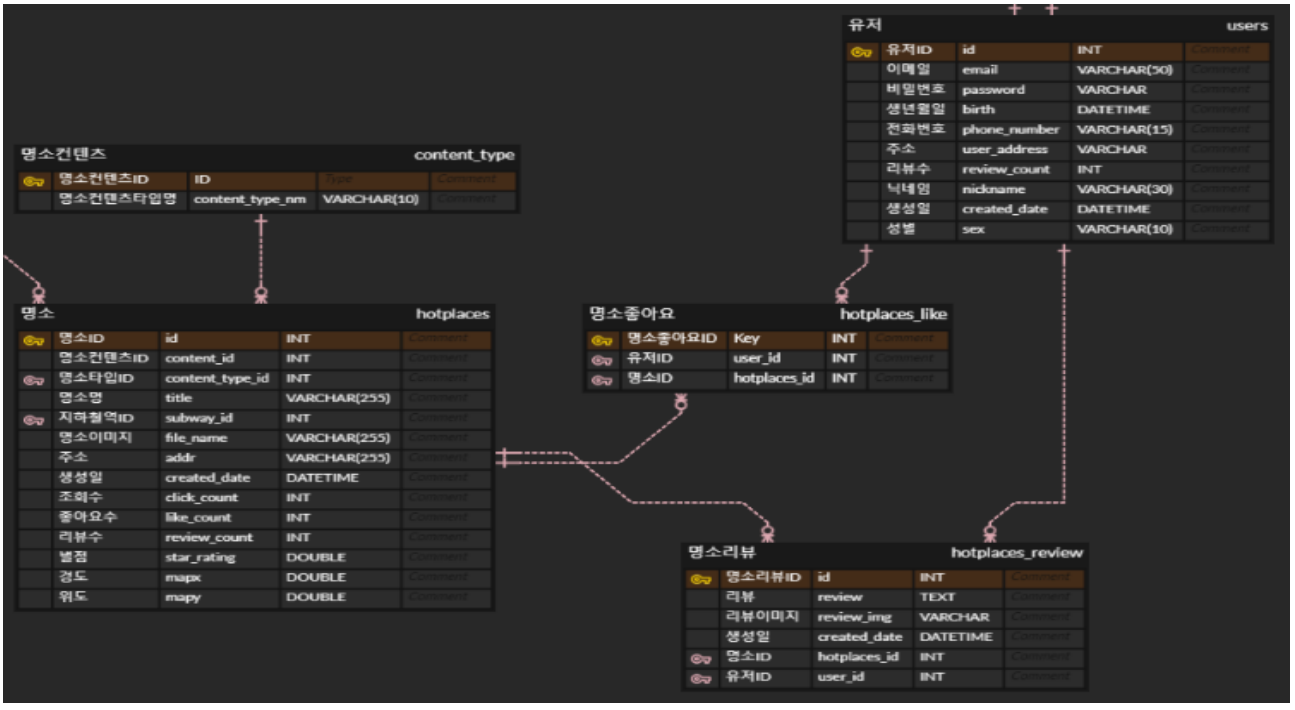


[부분 **:** 실시간 인구**,** 날씨 관련 테이블**]**

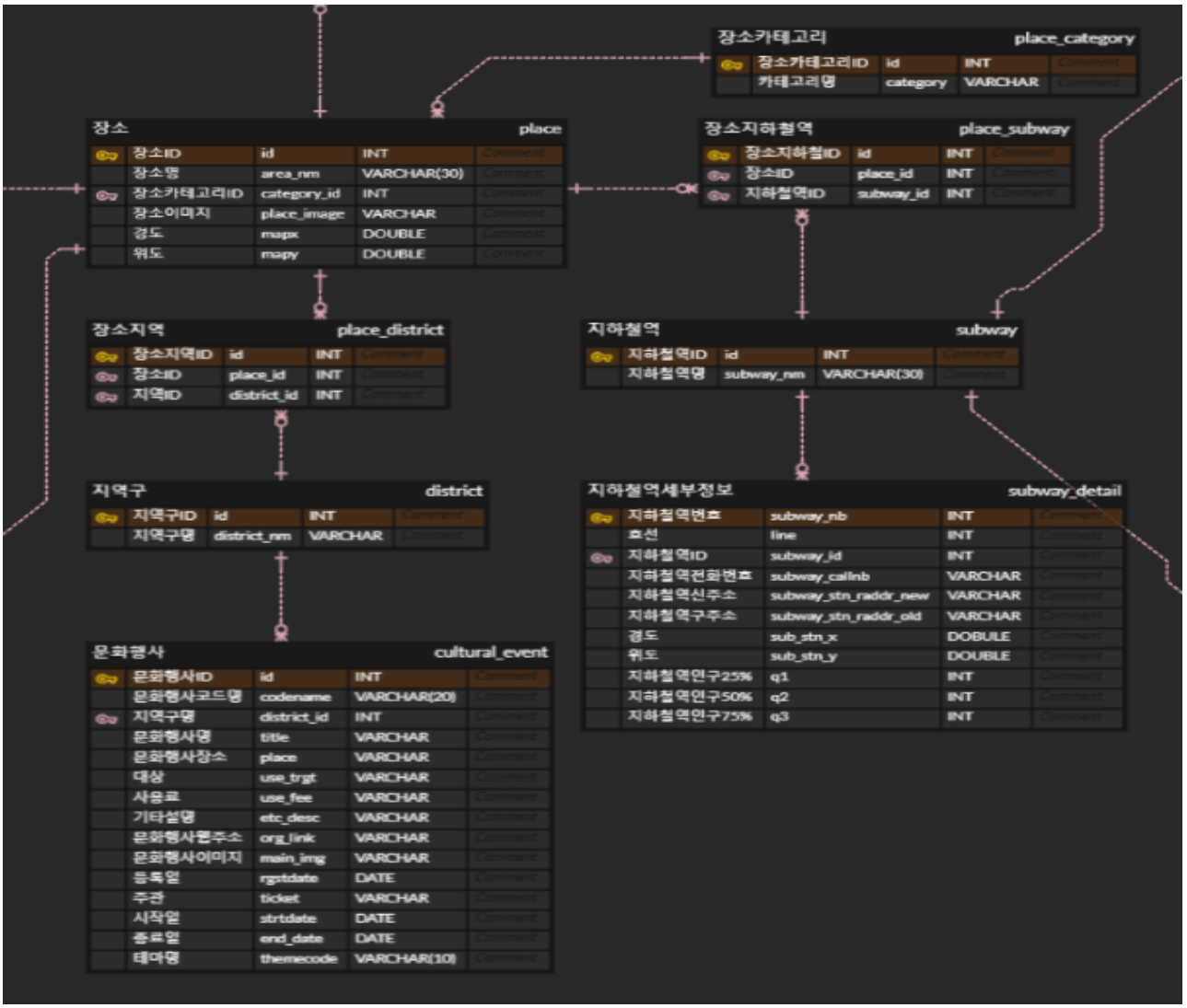
[부분 : 맛집 관련 테이블]



[부분 : 유저 및 명소 관련 테이블]



[부분 : 장소, 지역구, 지하철역 관련 테이블]

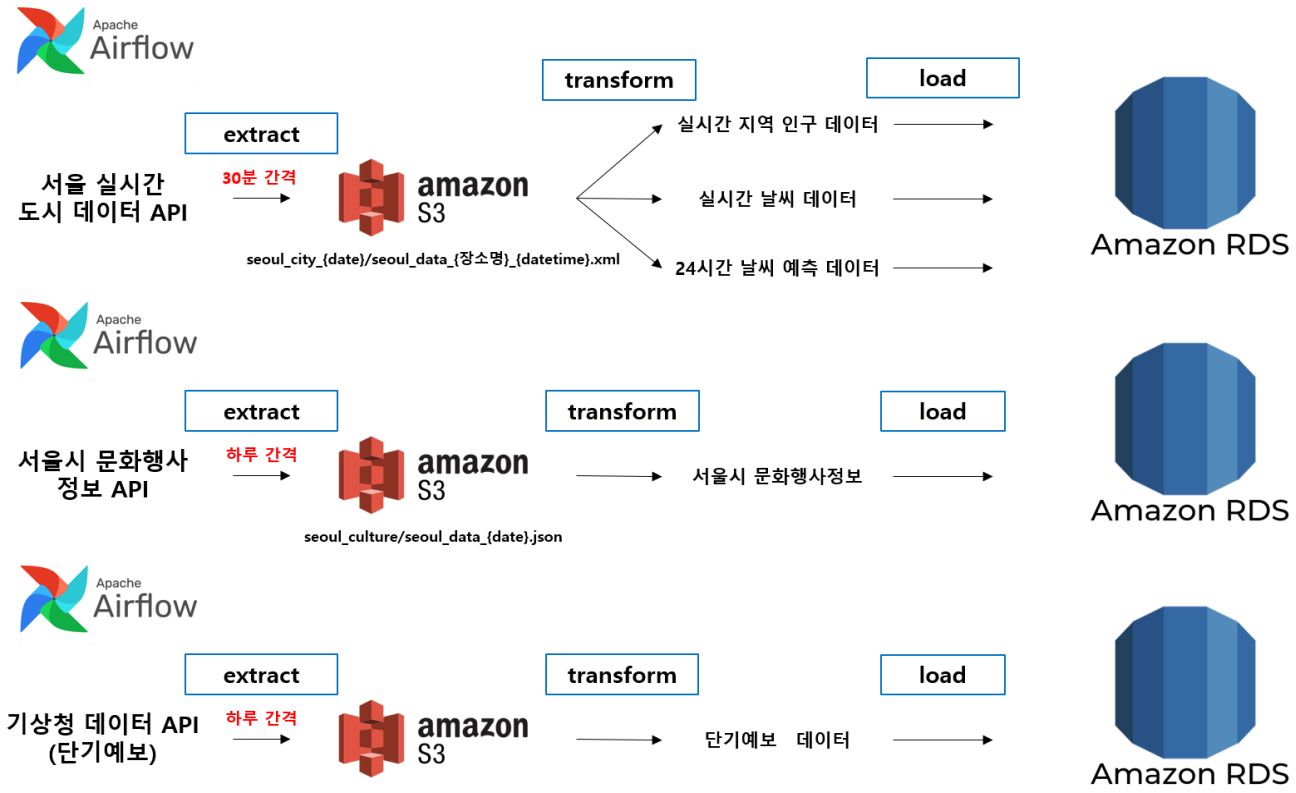


2.3. 데이터 파이프라인 설계

[운영용 파이프라인 설계 1]

해당 파이프라인은 실시간으로 변하는 서울시 인구, 날씨 정보와 하루 간격으로 변하는 문화행사 정보, 3일 간의 기상청 예보 정보를 운영 DB에 적재한다.

파이프라인명	스케줄링 주기	비고
서울시 실시간 도시데이터 파이프라인	매 30분 간격	분기 후 3 테이블 적재 - 실시간 지역 인구 - 실시간 날씨 - 24시간 날씨 예측
서울시 문화행사 정보 파이프라인	하루 간격	Full Refresh
기상청 데이터(단기예보) 파이프라인	하루 간격	Full Refresh

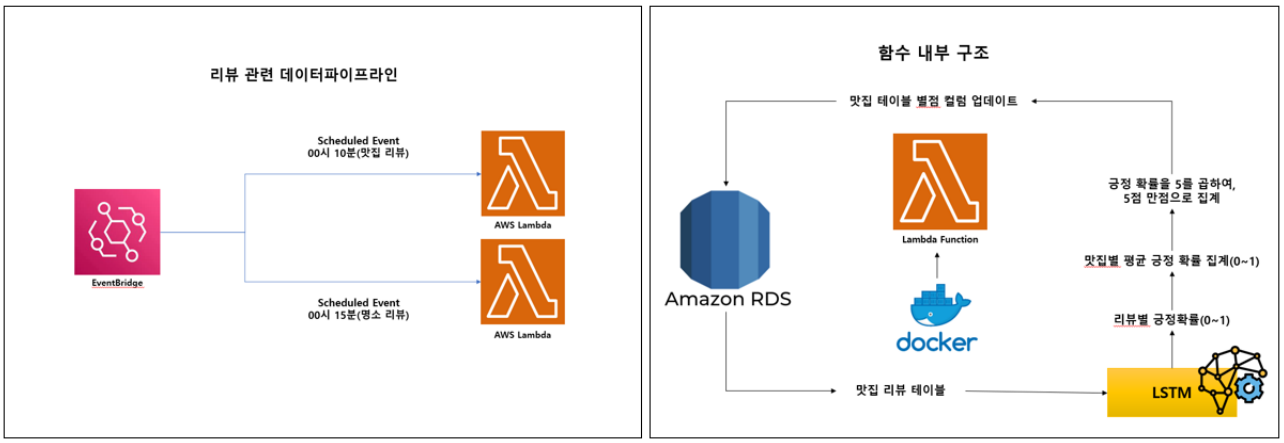


<운영용 파이프라인 1 도식화>

[운영용 파이프라인 설계 2]

해당 파이프라인은 맛집, 명소 관련 리뷰 데이터(자연어)를 기반으로 각 맛집, 명소별 별점(5점 만점)을 환산하여 운영 DB에 업데이트한다.

파이프라인명	스케줄링 주기	비고
리뷰 기반 별점 업데이트 파이프라인	하루 간격	딥러닝 모델 활용 (자연어 긍/부정 분류)



<운영용 파이프라인 2 도식화>

[분석용 파이프라인 설계 1]

해당 파이프라인은 운영용 DB에 적재된 정보를 하루 주기로 DW로 COPY 작업을 수행한다. 매일 업데이트 되는 정보를 데이터웨어하우스로 COPY하여 데이터 분석 및 시각화 과정에 사용한다.

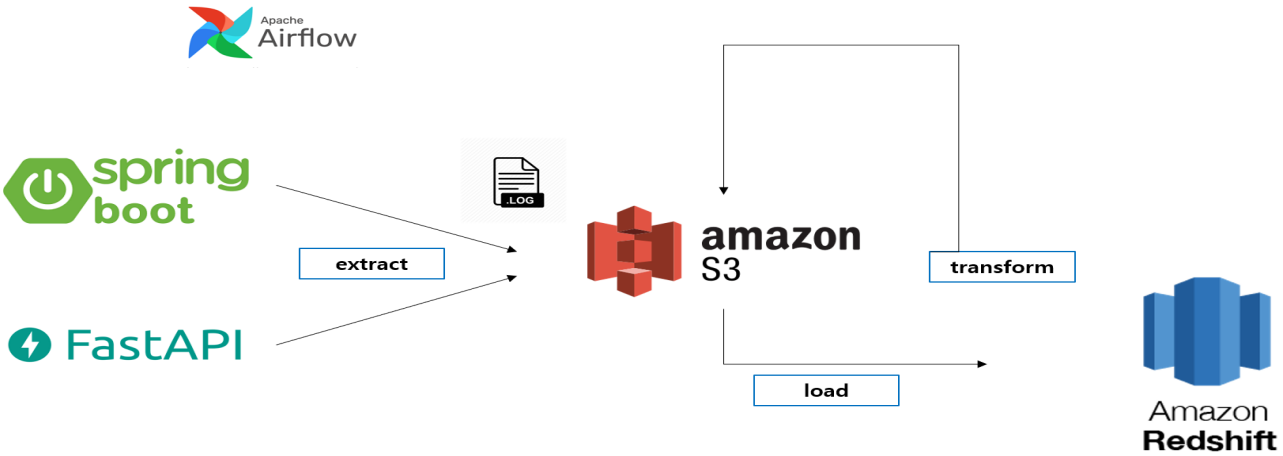
파이프라인명	스케줄링 주기	비고
DB 데이터 DW COPY 파이프라인	하루 간격	

<분석용 파이프라인 1 도식화>

[분석용 파이프라인 설계 2]

해당 파이프라인은 웹에서 발생한 로그 정보를 하루 주기로 DW에 정형 데이터 구조로 변환하여 적재한다. 매일 발생하는 로그 정보를 데이터 분석 및 시각화 과정에 사용한다.

파이프라인명	스케줄링 주기	비고
웹로그 가공 적재 파이프라인	하루 간격	

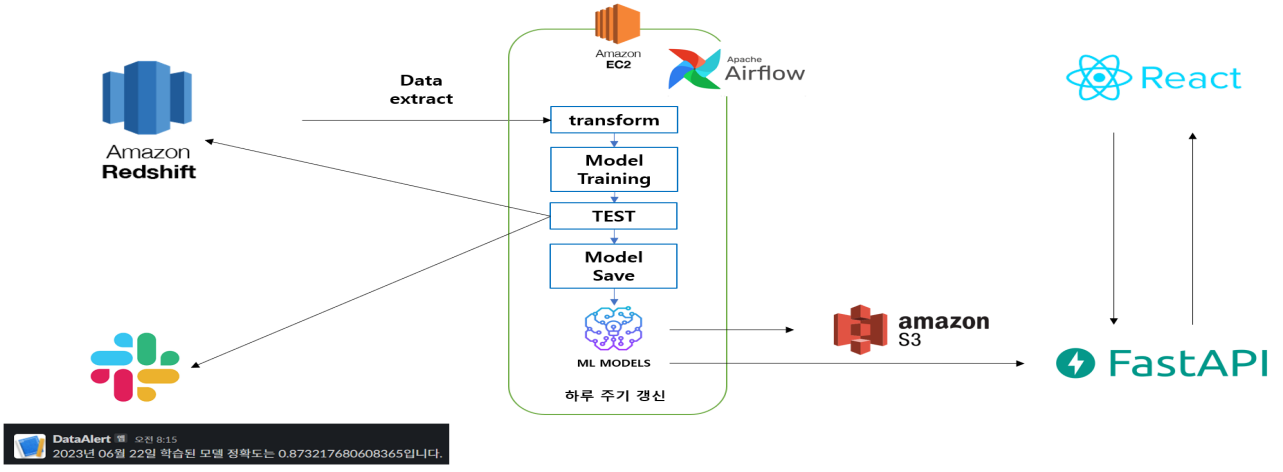


<분석용 파이프라인 2 도식화>

[ML 모델 자동화 배포 파이프라인]

해당 파이프라인은 DW에 적재된 장소별 인구, 날씨 데이터를 추출하고 변환한 뒤, 모델 학습 및 검증 과정 모니터링, 모델 저장 후 백엔드 서버(FASTAPI)로 전송하는 과정을 자동화한다.

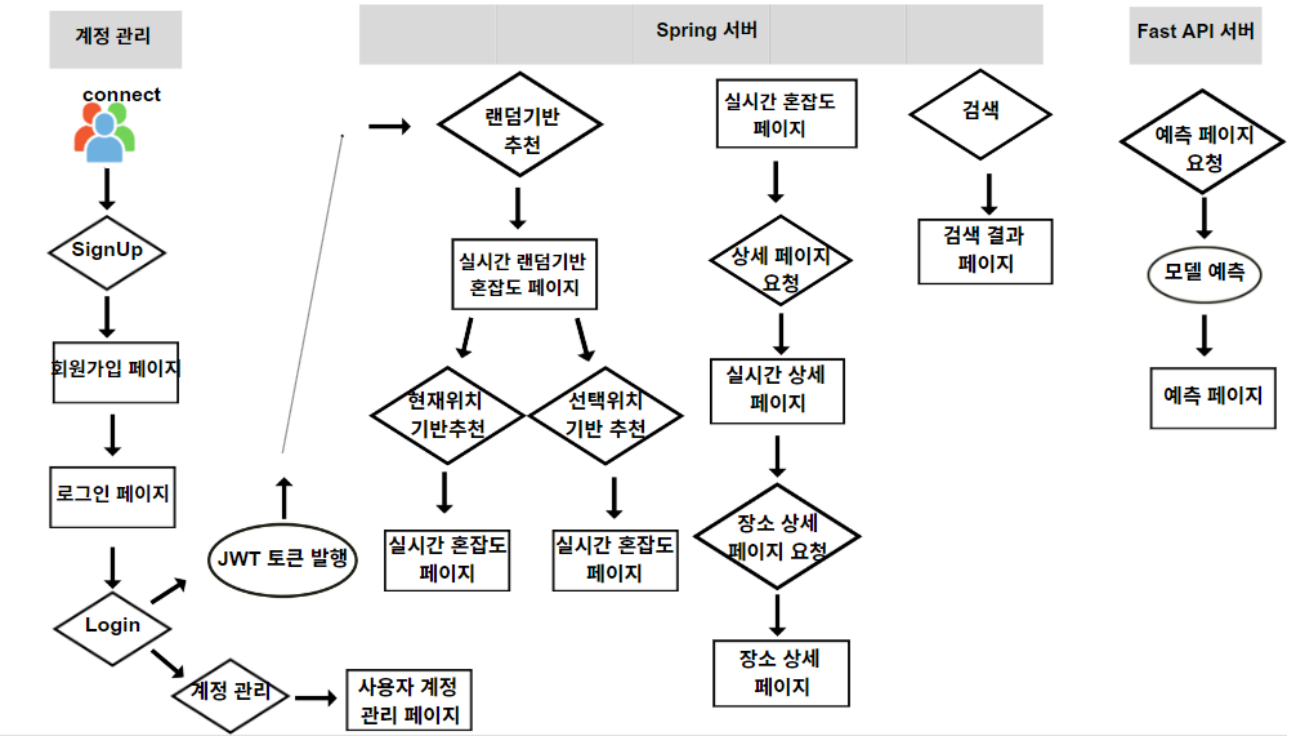
파이프라인명	스케줄링 주기	비고
ML 모델 자동화 배포 파이프라인	하루 간격	장소혼잡도 분류 모델 배포



<ML 모델 자동화 파이프라인 도식화>

2.4. 서비스 설계

[서비스 구성 장표]

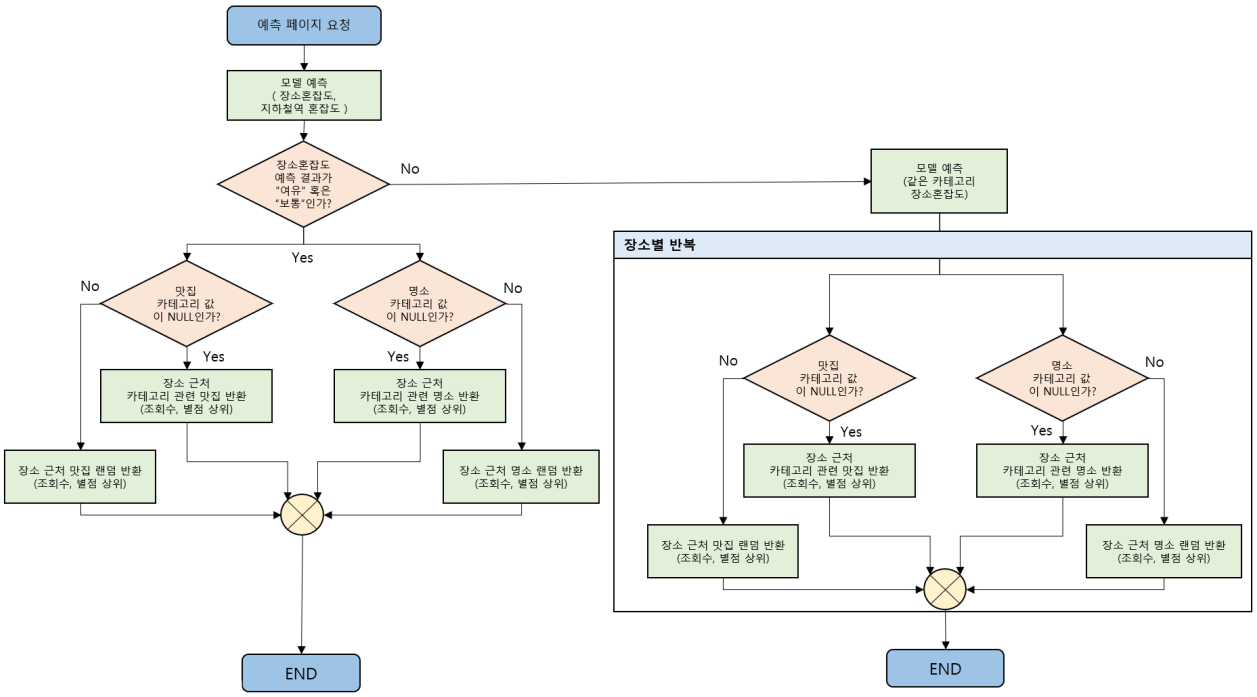


[서비스 구성 내용]

서비스명	내용
실시간 페이지	초기 접속시 '여유'인 장소를 랜덤 추천한다. 추후 사용자의 선택에 따라 현위치 or 지도에서 선택한 위치기반 '여유' 장소 추천
실시간 상세 페이지	모든 장소에 대해 '여유', '보통', '약간 붐빔', '붐빔'으로 분류하여 전체 장소를 보여준다
장소 상세 페이지	선택한 장소에 대한 1주일 통계 대시보드 제공 및 현재 혼잡도 여부 제공. 혼잡여부에 따라 식당, 놀거리, 문화행사 추천 또는 다른 '여유' 장소추천.

예측 페이지	**1.** 사용자에게 날짜, 시간, 장소를 입력받아 ML 모델을 활용한 해당 시간대, 특정 장소의 혼잡도 분류 **2.** 혼잡도에 따른 추천 a. 혼잡도 '여유', '보통'일 때 i. 해당 장소 및 사용자 선택 카테고리 맛집, 명소 추천 ii. 인근 지하철역 혼잡도 예측 정보 제공(ML) b. 혼잡도 '약간 붐빔', '붐빔'일 때 i. 같은 카테고리로 분류된 다른 장소의 사용자 선택 카테고리 맛집, 명소 추천 ii. 인근 지하철역 혼잡도 예측 정보 제공(ML) **3.** **(**추천 알고리즘 상세**)** a. 조회수, 좋아요 수, 딥러닝 모델로 집계된 별점을 활용한 상위 맛집, 명소 추천
로그인, 회원가입	로그인 시 JWT 토큰을 발행하여, 해당 토큰이 없다면 사이트 이용이 불가능하게끔 제어.
추천 알고리즘	댓글, 좋아요, 조회수 데이터를 통해 해당 장소에 대한 가중치 증가를 통한 추천 확률 증가
로그 저장	Fast API, Spring 서버에서 Controller에 대한 GET,POST와 같은 요청이 발생할 경우 Interceptor layer에서 Info Level에 대한 로그를 저장. 서버에서 저장된 로그 파일을 Airflow에서 호출하면 AWS s3에 저장.

[예측 페이지 알고리즘 상세**]**



2.5. 모델 설계

전체 모델링 로드맵

데이터 수집 -> 데이터 전처리 -> EDA -> 변수 선택 -> 파생 변수 생성 -> 모델선택 -> 모델 최적화 -> 모델 평가

1. 지하철 승하차 인원 예측 모델 설계

모델 기본 설계 **(Model Design)**

Model : Lasso, Ridge, ElasticNet, CatBoostRegressor, XGBRegressor, LGBMRegressor

Scaler : Log Scaler

Target : 승하차 인원

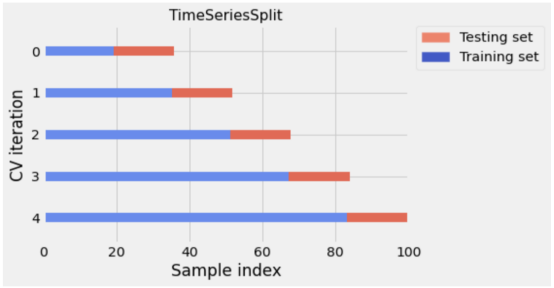
Feature : 역 번호, 호선, 연도, 월, 시간, 요일, 공휴일여부, 날씨데이터

Split : Train Data(2018~2021 승하차인원 데이터), Test Data(2022~2023.4 승하차인원 데이터)

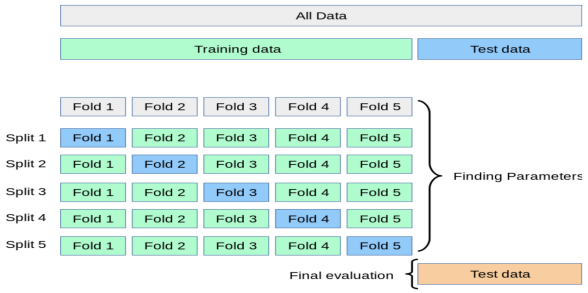
모델을 통하여 특정 날짜, 날씨에 대해 지하철 승하차 인원을 예측을 하여 사람이 많은 시간을 피해 지하철 이용의 불편함을 해소하고자 하며, 해당 역에 인원이 많을시 다른 근처 역을 알려줌.

2018년도부터 2021년도까지의 지하철 승하차 데이터를 훈련 데이터로 사용하고, 2022년도부터 2023년도까지의 데이터를 테스트 데이터로 사용함. 이를 통하여 모델은 과거 데이터를 기반으로 학습하여 미래에 데이터에 대해 예측 수행.

Nested Cross validation를 사용하여, 외부 시계열 교차 검증과 내부 모델 검증을 통하여 모델의 일반화 성능 확인을 가능하게 합니다.



[외부 시계열 교차 검증]



[내부 모델 검증]

외부 시계열 교차 검증

- 시계열 데이터에는 데이터 포인트의 순서가 중요한 시간적 순서가 있습니다. 기존의 교차 유효성 검사에서는 데이터가 무작위로 섞이고 폴드로 분할되어 향후 정보가 학습 프로세스로 유출될 수 있습니다.
- 시계열 교차 검증은 모델 평가 중에 데이터의 시간적 순서를 보존하여 이 문제를 해결하도록 설계되었습니다.
- 이렇게 하면 모델이 과거 데이터에 대해 교육되고 미래 데이터에 대해 평가될 수 있으므로 모델이 과거 데이터에 대해 교육되고 미래 데이터에 대해 테스트되는 실제 시나리오를 시뮬레이션합니다.

내부 모델 검증

- 시계열 교차 검증의 각 폴드 내에서 내부 루프는 훈련 및 검증 세트를 사용하여 모델을 훈련하고 평가합니다.

[평가 메트릭]

실제 승차인원과 예측 승차인원 차의 평균을 직관적으로 확인하고 특이값이 많아 모델의 평가 기준을 MAE(Mean Absolute Error)로 설정 하게 되었습니다.

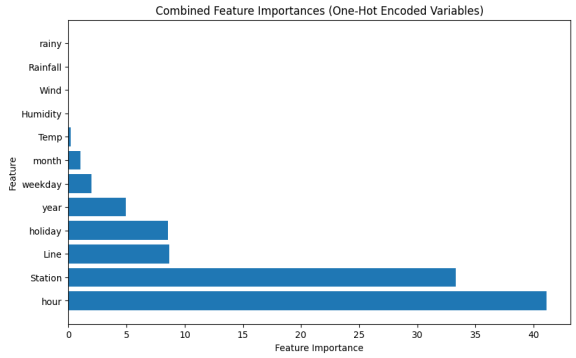
[모델 후보]

성능 비교를 통해 상위 3개의 모델 CatBoost, XGBoost, LGBM을 모델 후보로 선정 하게 되었습니다.

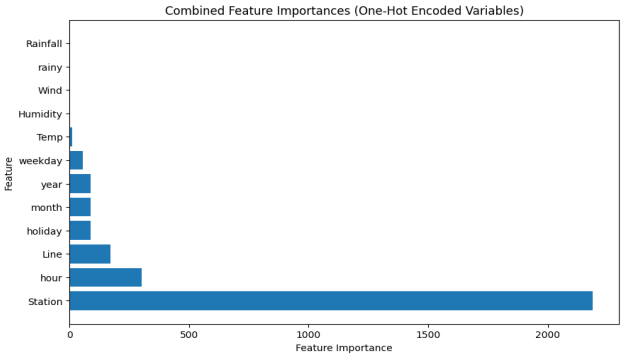
	Model	MAE (CV)	MAE (Train)	MAE (Test)	R2 Score (Train)	R2 Score (Test)
0	Lasso	1211.498289	1186.132463	1000.049655	-0.120831	-0.078943
1	Ridge	758.926517	768.555694	733.085378	0.499207	0.378390
2	ElasticNet	1209.339334	1183.340583	996.901810	-0.118006	-0.081600
3	CatBoostRegressor	257.113899	234.421306	390.495542	0.932182	0.818761
4	XGBRegressor	422.720453	399.179642	394.115101	0.847583	0.822342
5	LGBMRegressor	412.705568	397.045196	430.061633	0.839545	0.782966

[모델 선정**]**

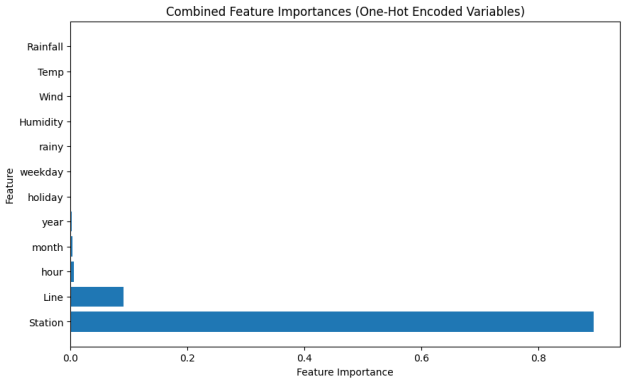
상위 3개의 모델의 특성 중요도를 비교 하였을때 특성 중요도를 가장 골고루 판단하고, 시간과 역을 가장 높게 평가한 CatBoost 모델을 선정하게 되었습니다.



[CatBoost 특성 중요도]



[LGBM 특성 중요도]



[XGBoost 특성 중요도]

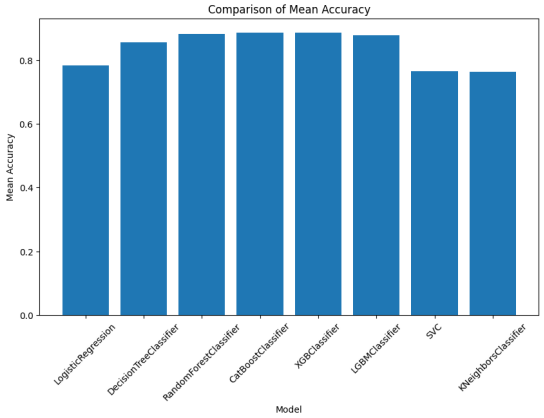
2. 장소 혼잡도 분류 모델 설계

모델 기본 설계 ***(Model Design)***

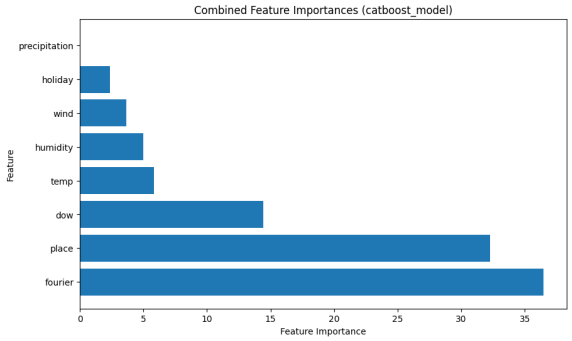
Model : LogisticRegression, DecisionTreeClassifier,RandomForestClassifier, CatBoostClassifier,XGBClassifier,LGBMClassifier,SVC, KNeighborsClassifier
Scaler : 사용 안함
Target : 장소 혼잡도(0~3)
Feature : 장소, 월, 시간, 요일, 공휴일여부, 날씨데이터(온도, 습도, 풍속, 강수량)

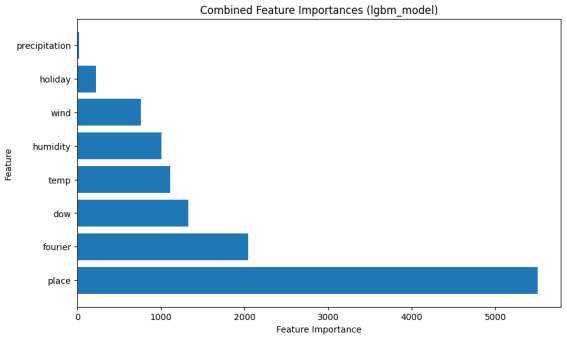
특정 날짜의 장소, 요일, 공휴일, 온도, 습도, 풍속, 강수량, 시간대 특성을 이용하여, 장소의 혼잡도를 분류하는 ML 기반 분류 모델. 2023년 6월 초부터 DB에 적재된 데이터 기반으로, 학습하였으며, 매일 적재되는 데이터를 이용하여, 매일 자동으로 배포되는 모델

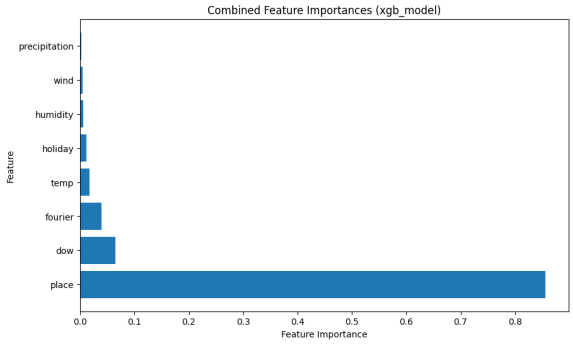
[모델 후보]
평가 메트릭을 정확도로 사용하여 ,성능 비교를 통한 상위 3개의 모델 CatBoost, XGBoost, LGBM을 모델 후보로 선정 하게 되었습니다.



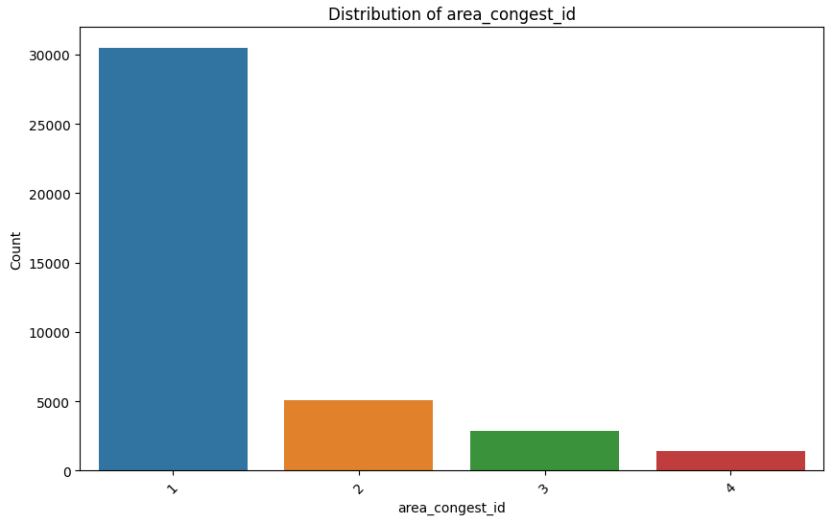
[모델 선정]
- 이 중 CATBOOST는 변수의 영향력을 골고루 판단하고, 특히 fourier 관련 변수의 중요도를 높게 선정하여 시간과 관련된 장소 혼잡도 예측 결과를 유의미하게 다뤘다고 판단하였습니다.







[클래스 불균형**]**



SMOTE 기법과, 클래스별 가중치 부여 중에 고민하였다. SMOTE 기법을 적용한 결과 과적합 문제와, 새로운 데이터에 대한 예측력이 현저히 떨어지는 문제로 인하여, 클래스별 가중치를 적용하였다. 클래스별 가중치는, 각 클래스 간 비율의 역으로 계산하였다.

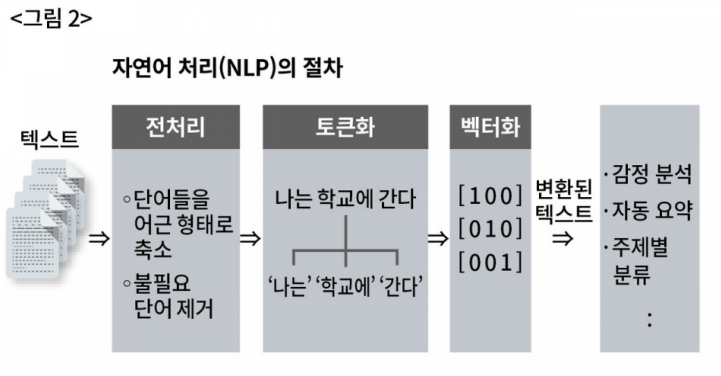
[하이퍼 파라미터 튜닝**](**수정 예정**)**

Best Parameters: {'bagging_temperature': 0.5, 'depth': 7, 'iterations': 1000, 'l2_leaf_reg': 1, 'learning_rate': 0.5, 'random_strength': 0.5}

3. 자연어 기반 긍**/**부정 분류 모델 설계

자연어 처리 절차

<그림 2>



(1) 데이터 전처리 : 중복값, null값 제거, 한글과 공백을 제외하고 모두 제거, 한글만 있는 데이터 정제, 불용어제거, 패딩 처리
(2) 토큰화 : 텍스트를 개별 단어로 분리.
(3) 벡터화 : 문자로 되어있는 리뷰데이터를 인덱스 벡터화 진행.

3. 구현

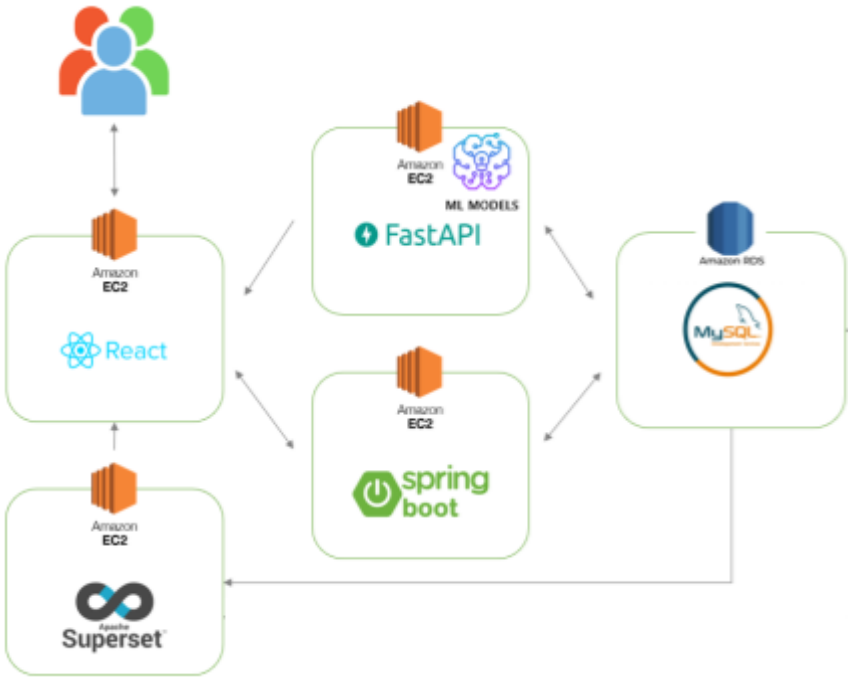
3.1. 개발 환경

- 개발 언어 : Python, Java, JavaScript, HTML, CSS
- 프론트엔드 : HTML, CSS, ReactJS
- 백엔드 : Spring, SpringBoot, FastAPI
- 데이터 : Tensorflow, Scikit-Learn, Airflow, Pandas, Spark, Superset
- 데이터베이스 : MySQL(AWS RDS), AWS Redshift
- 테스팅툴 : Selenium, Postman
- 클라우드 : AWS(EC2, S3, RDS, Redshift, ECR, Lambda)
- 데브옵스 : Git, GitHub, GitKraken, Docker
- 협업툴 : Notion, Slack, Canva

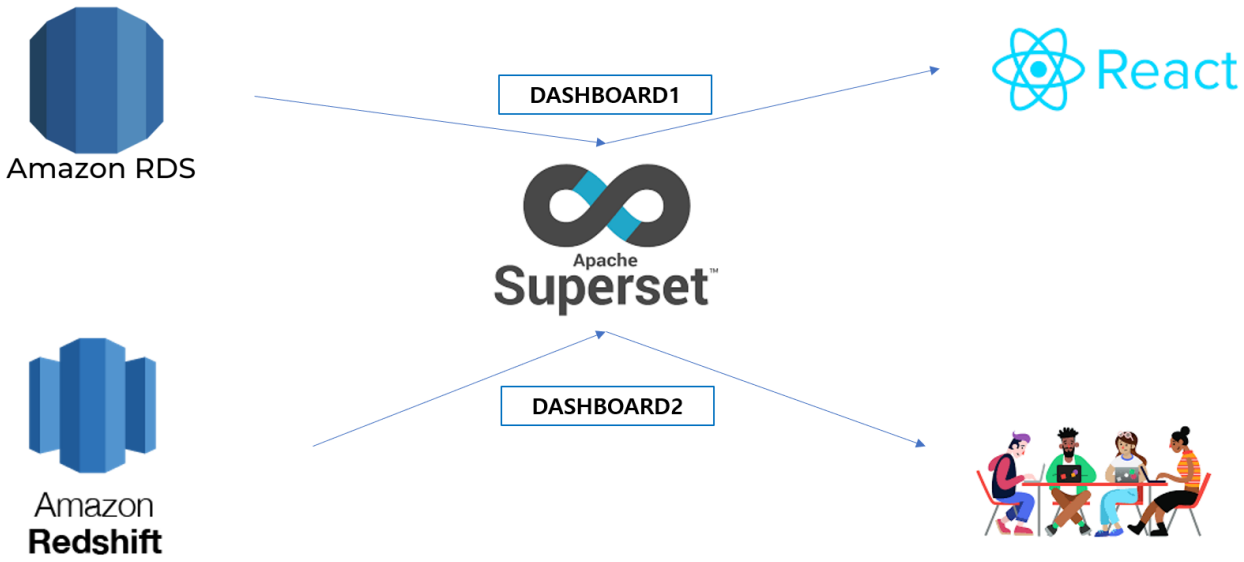
3.2. 배포 환경

- AWS를 통한 클라우드 환경 진행

서버명	**Service tier**	**vCpus**	**Memory**	**OS**
Airflow 전용 서버	t3.medium	2	4GiB	Ubuntu 22.04
Superset 전용 서버	t3.small	1	2GiB	Ubuntu 22.04
Fastapi 전용 서버	t3.small	1	2GiB	Ubuntu 22.04
Web 전용 서버	t3.small	1	2GiB	Ubuntu 22.04
RDS(MYSQL)	t2.micro	1	1GiB	


Amazon
EC2
ML MODELS
FastAPI
Amazon
EC2
React
Amazon RDS
MySQL
Amazon
EC2
spring
boot
Amazon
EC2
Apache
Superset

[대시보드 활용]


Amazon RDS
DASHBOARD1
React
Apache
Superset
DASHBOARD2
Amazon
Redshift

3.3. 데이터 출처

- 서울교통공사_역별 일별 시간대별 승하차인원 정보(CSV)
- 서울시 실시간 도시데이터(주요 48 장소)(OPEN API)
- 서울시 문화행사 정보(OPEN API)
- 기상청 단기예보(OPEN API)
- 서울시 주요 48 장소 맛집 크롤링
- 한국관광공사 명소 데이터(OPEN API)
- 날씨 데이터(CSV)
- 서울시 주요 50장소 지하철역 매핑 데이터