Data Science / EDA / Machine Learning Project

플레이스투밋(P2M) – 이민호 | 정재영 | 이서영

교통, 상권, 날씨를 고려한

혼잡 여부 예측 모델 개발

교통, 상권, 날씨를 고려한

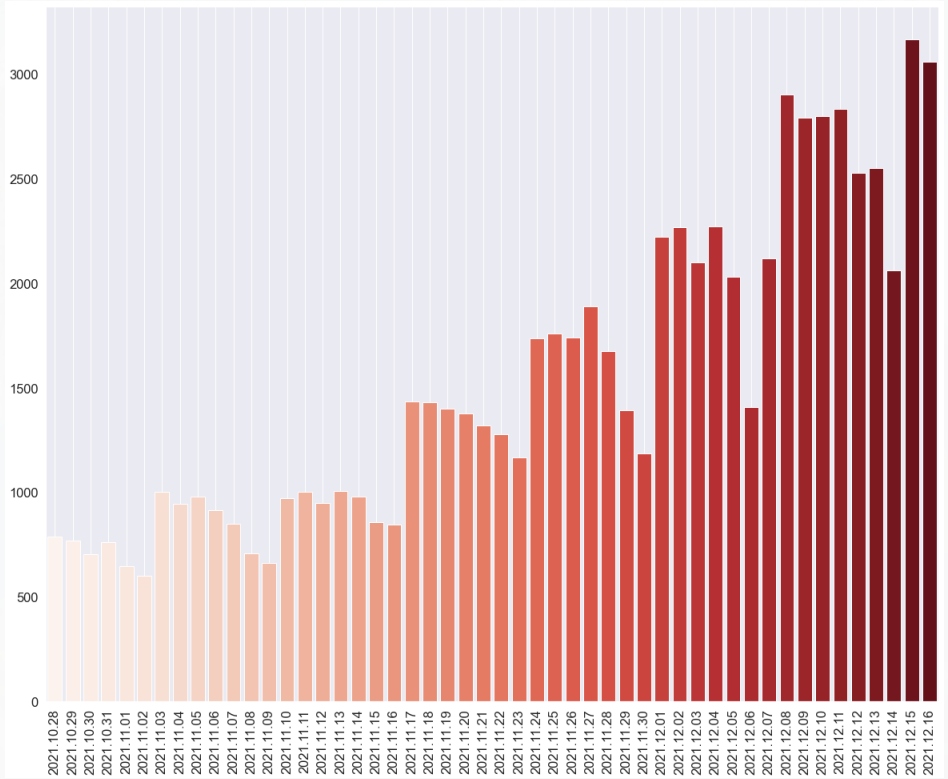
혼잡 여부 예측 모델 개발 이렇게 진행합니다.

프로젝트 소개	탐색적 데이터 분석 (EDA)	모델링	프로젝트 결과	마무리
Goal & Data	Preprocessing & Visualization	ML Modeling	Result & Comparision	Review

1. 프로젝트 소개

매일 같은 고민, “어디서 만나지?”

WHO	WHEN	WHAT
너와 나랑	저녁 7시	밥 혹은 술을
HOW	**WHY**	**WHERE**
분위기 있게	불금이잖아	몰라



서울시 코로나19 신규 확진자 수 추이 (2021-10-28 이후)
- 데이터 : 서울 열린데이터 광장 (발생동향)

최근 코로나 19 위험 지역 등 고려할 것들이 많아, **약속장소를 정하는 일이 쉽지 않다**.

우리가 피하고자 하는 곳은 어디인가?

“혼잡한 곳”

상권이 많이 위치한 곳

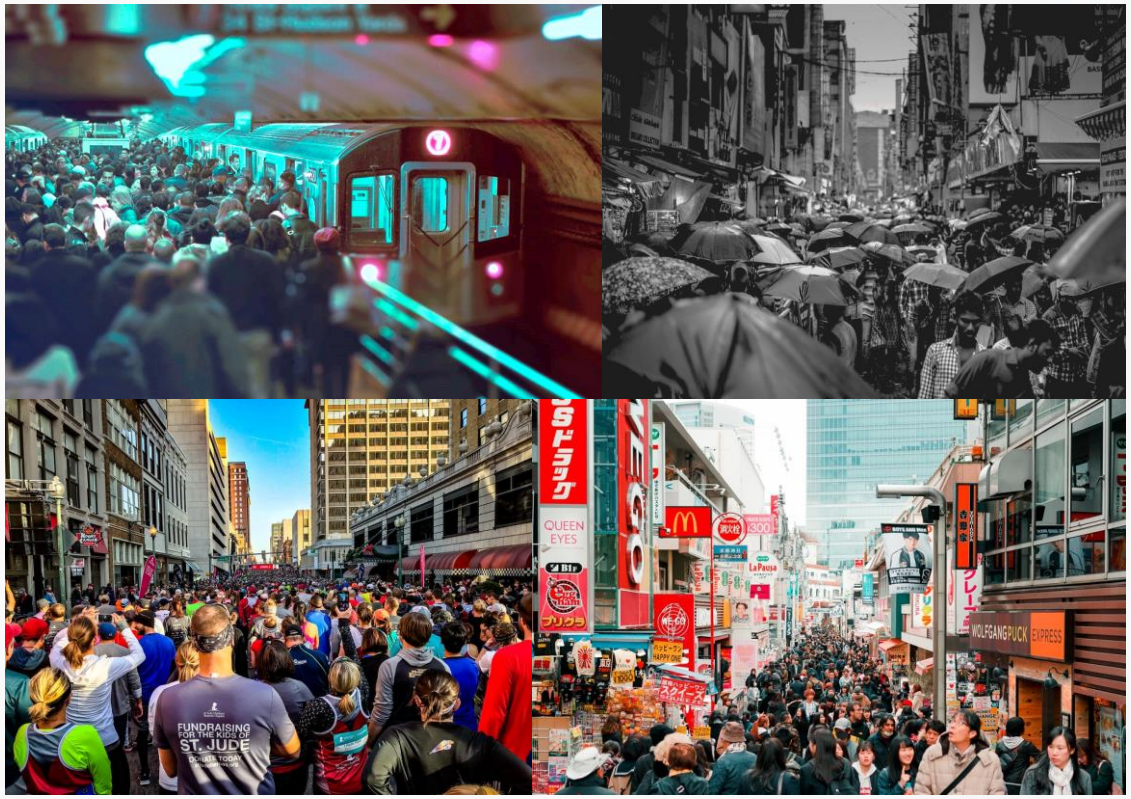
볼거리가 많은 곳

비가 오면 실내

날 좋으면 실외

퇴근하면 그렇게 멀지 않은 곳

…



혼잡 정도와 관련한 여러 요소들을 고려하여, 약속 시간 부근에 가려고 하는 곳이 **혼잡한지 예측**할 필요가 있다.

프로젝트 개요

프로젝트 목표

내가 가려는 지하철역 인근이 혼잡할 지 예측한다.

혼잡 : 1
비혼잡 : 0

← Labeling

지하철역 하차인원

← ML Modeling

Feature

지하철명, 요일, 날씨, 인구수
관광지수, 상권수, 사업체수

***프로젝트 전제 조건:**

1. 코로나에 따른 변화를 고려하여, 데이터는 2019-2020년을 기준으로 프로젝트를 수행
2. '약속 장소 선정' 콘셉트에 맞게, 일별 오후 6시 ~ 8시 데이터를 기준으로 진행
3. 해당 인근의 혼잡한 정도는 하차인원과 비례한다고 가정
4. 서울권 내 지하철 가운데 1~8호선으로 한정

프로젝트 상세 내용

1. **프로젝트명** : 교통, 상권, 날씨를 고려한 혼잡 여부 예측 모델 개발
2. **수행자** : 플레이스투밋(P2M) – 이민호, 정재영, 이서영
3. **수행기간** : 3주 (2021.12월 중)
4. **목표** : 지하철역 기준 인근 지역의 혼잡 여부 및 해당 확률 제시
5. **구성**
   - 데이터 전처리 : 2019, 2020년(2개년) 저녁 6시 ~ 8시 데이터
   - 학습 데이터 혼잡 여부 라벨링 : (산출식 〉= 1.3) 기준
   - 탐색적 데이터 분석 : EDA 및 시각화
   - 혼잡 여부 예측 모델링 : 분류(Classification)
6. **데이터셋**

구분	내용	활용 정보
지하철역	서울시 호선별 지하철역 정보	전철역 / 호선 / 역 코드
	서울시 지하철 호선별 역별 시간대별 승하차 인원 정보	역별 / 시간대별 / 승차인원 / 하차인원
상권 정보	지하철 노선별 관광지	호선 / 지하철역 / 관광지명 / 주소 등
	소상공인시장진흥공단_상가(상권)정보_20201231_서울	사업체명 / 표준산업분류코드 / 표준산업분류명 / 행정동
	서울시 사업체현황 (조직형태별/동별) 통계_2019	행정동별 산업분류에 따른 사업체 수
	서울시 우리마을가게 상권분석서비스(상권영역)	상권구분 / 행정동코드
날씨	2개년도 일별 시간대별 날씨 데이터	일시 / 기온 / 풍속 / 적설 / 강수 정보
인구수	서울시 주민등록인구 (동별) 통계	동별 인구수

2. EDA

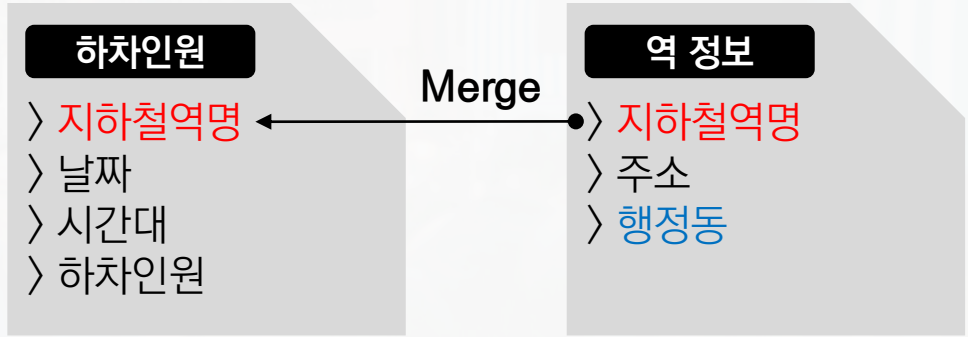
1) 데이터 전처리 (Data Preprocessing)

[데이터셋]

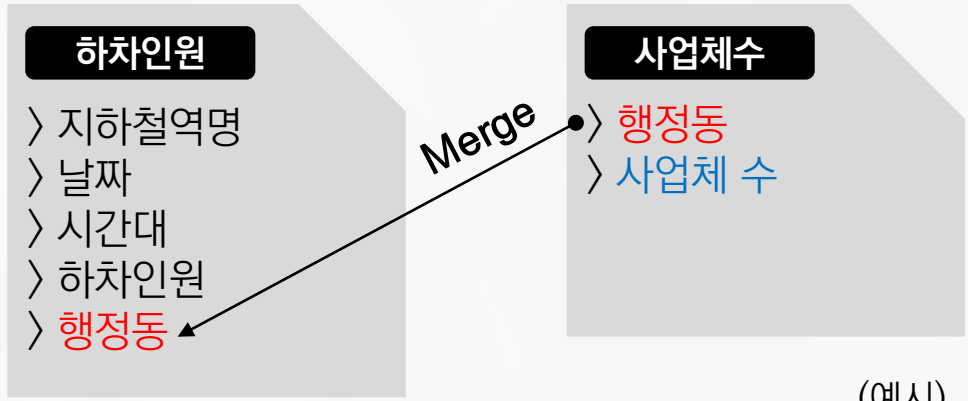
- 일별 시간대별 하차인원 데이터 **하차인원**
- 지하철역(1~8호선) 정보 데이터 **역 정보**
- 지하철역 인근 관광지 데이터 **관광지수**
- 서울시 사업체 현황 데이터 **사업체수**
- 일별 시간대별 날씨 데이터 **날씨**
- 서울시 주민등록인구 통계 데이터 **인구수**
- 소상공인시장진흥공단 상권 정보 데이터 **상권수**

하차인원 데이터 기준으로 Merge하는 방식 (how='left')

1〉



2〉



(예시)

1) 데이터 전처리 (Data Preprocessing) – 계속

A. 2020년 사업체 수 데이터 부재 Issue **사업체수**

✓ **2019년 행정동별 사업체 수 데이터** : 업종별 행정동별 사업체 수가 정리되어 있음

자치구	동	합계					농업 임업 및 어업		광업		제조업		가스 증기 및 공기조절 공급업		수도 및 폐기물 처리 원료		건설업	
		사업체수		종사자수			사업체수	종사자수	사업체수	종사자수	사업체수	종사자수	사업체수	종사자수	사업체수	종사자수	사업체수	종사자
		사업체수	여성대표자	계	남	여												
합계		822,863	278,152	5,119,913	2,848,150	2,271,763	29	469	21	66	61,583	277,920	139	5,576	445	8,963	22,228	385,2
	소계	40,490	13,577	268,702	154,100	114,602	3	30	4	10	4,450	13,078	7	363	6	136	339	30,75
	사직동	3,544	1,147	52,489	31,029	21,460	1	16	-	-	72	217	1	8	2	31	37	7,33
	삼청동	753	360	4,507	2,225	2,282	-	-	-	-	16	82	-	-	-	-	8	42
	부암동	567	235	4,147	2,137	2,010	-	-	1	5	24	122	-	-	-	-	23	62
	평창동	739	327	3,547	1,679	1,868	-	-	-	-	15	101	-	-	-	-	23	98
	무악동	531	104	1,457	729	728	-	-	-	-	6	11	-	-	-	-	8	55
	교남동	347	131	3,816	1,553	2,263	-	-	-	-	7	15	-	-	-	-	9	52
	가회동	747	366	13,681	10,225	3,456	-	-	-	-	33	287	-	-	-	-	12	7,89
	종로1.2.3.4가동	15,851	5,042	111,270	66,889	44,381	-	-	3	5	1,927	5,316	5	353	2	18	78	14,26

✓ **2020년 서울시 내 사업체 정보 Raw 데이터** : 2020년 정리된 데이터 없어서 별도로 전처리 수행

상가업소번	상호명	지점명	상권업종대	상권업종대	상권업종중	상권업종중	상권업종소	상권업종소	표준산업분	표준산업분	시도코드	시도명	시군구코드	시군구명	행정동코드	행정동명
17163092	도전최강달인왕만두		Q	음식	Q01	한식	Q01A01	한식/백반/	I56111	한식 음식점	11	서울특별시	11740	강동구	1.17E+09	고덕2동
17120456	이때		Q	음식	Q01	한식	Q01A01	한식/백반/	I56111	한식 음식점	11	서울특별시	11440	마포구	1.14E+09	연남동
17175350	L.A.D		Q	음식	Q01	한식	Q01A01	한식/백반/	I56111	한식 음식점	11	서울특별시	11440	마포구	1.14E+09	서교동
17175311	제이씨에스푸드		Q	음식	Q01	한식	Q01A01	한식/백반/	I56111	한식 음식점	11	서울특별시	11620	관악구	1.16E+09	서림동
22767534	BYC상신점	상신점	D	소매	D05	의복의류	D05A07	셔츠/내의/속옷			11	서울특별시	11710	송파구	1.17E+09	거여2동
25258802	윤선생어학원		R	학문/교육	R01	학원-보습	R01A01	학원-입시	P85501	일반 교과	11	서울특별시	11560	영등포구	1.16E+09	영등포본동
23324279	제중건강원		D	소매	D10	건강/미용	D10A07	건강원	G47216	건강보조식	11	서울특별시	11560	영등포구	1.16E+09	영등포동
24525909	민속악기사		D	소매	D04	취미/오락	D04A09	악기판매	G47593	악기 소매업	11	서울특별시	11200	성동구	1.12E+09	용답동
23958280	미리어케이		F	생활서비스	F13	기타서비스	F13A13	상품전시	N75992	전시 및 행	11	서울특별시	11680	강남구	1.17E+09	역삼1동
15554136	김성희꼼꼼국어교		R	학문/교육	R01	학원-보습	R01A01	학원-입시	P85501	일반 교과	11	서울특별시	11710	송파구	1.17E+09	방이1동

2019년 포맷으로 전처리 수행

1) 데이터 전처리 (Data Preprocessing) – 계속

B. 행정동 통일 Issue

인구수 | 사업체수 | 상권수

```
people.행정동.unique()

array(['사직동', '삼청동', '부암동', '평창동', '무악동', '교남동', '가회동', '종로1.2.3.4가동',
       '종로5·6가동', '이화동', '창신1동', '창신2동', '창신3동', '숭인1동', '숭인2동', '청운효자동',
       '혜화동', '소계', '소공동', '회현동', '명동', '필동', '장충동', '광희동', '을지로동',
       '신당5동', '황학동', '중림동', '신당동', '다산동', '약수동', '청구동', '동화동', '후암동',
       '용산2가동', '남영동', '원효로2동', '효창동', '용문동', '이촌1동', '이촌2동', '이태원1동',
       '이태원2동', '서빙고동', '보광동', '청파동', '원효로1동', '한강로동', '한남동', '왕십리2동',
       '마장동', '사근동', '행당1동', '행당2동', '응봉동', '금호1가동', '금호4가동', '성수1가1동',
       '성수1가2동', '성수2가1동', '성수2가3동', '송정동', '용답동', '왕십리도선동', '금호2.3가동',
       '옥수동', '화양동', '군자동', '중곡1동', '중곡2동', '중곡3동', '중곡4동', '능동', '구의1동',
       '구의2동', '구의3동', '광장동', '자양1동', '자양2동', '자양3동', '자양4동', '회기동',
       '휘경1동', '휘경2동', '청량리동', '용신동', '제기동', '전농1동', '전농2동', '답십리1동',
       '답십리2동', '장안1동', '장안2동', '이문1동', '이문2동', '면목2동', '면목4동', '면목5동',
       '면목7동', '상봉1동', '상봉2동', '중화1동', '중화2동', '묵1동', '묵2동', '망우3동',
       '신내1동', '신내2동', '면목본동', '면목3.8동', '망우본동', '돈암1동', '돈암2동', '안암동',
       '보문동', '정릉1동', '정릉2동', '정릉3동', '정릉4동', '길음1동', '길음2동', '월곡1동',
       '월곡2동', '장위1동', '장위2동', '장위3동', '성북동', '삼선동', '동선동', '종암동', '석관동',
       '번1동', '번2동', '번3동', '수유1동', '수유2동', '수유3동', '삼양동', '미아동', '송중동',
       '송천동', '삼각산동', '우이동', '인수동', '쌍문1동', '쌍문2동', '쌍문3동', '쌍문4동',
       '방학1동', '방학2동', '방학3동', '창1동', '창2동', '창3동', '창4동', '창5동', '도봉1동',
       '도봉2동', '월계1동', '월계2동', '월계3동', '공릉2동', '하계1동', '하계2동', '중계본동',
       '중계1동', '중계4동', '상계1동', '상계2동', '상계5동', '상계8동', '상계9동', '상계10동',
       '상계3.4동', '상계6.7동', '중계2.3동', '공릉1동', '녹번동', '불광1동', '갈현1동',
```

인구수 데이터의 행정동 컬럼의 unique

- ‘– 동’ : 기본형
- ‘– 1동’ : 1차 변형
- ‘– 1.2동’ : 1차 변형 응용
- ‘– 1가동’ : 2차 변형
- ‘– 1.2가동’ : 2차 변형 응용
- ‘– 1.2가1동’ : 2차 변형 응용2

> ‘– 동’ (기본형)

묶여 있는 행정동의 각 데이터를 분리시킬 수 없기 때문에, 기본형(–동)으로 모두 병합·통일

1) 데이터 전처리 (Data Preprocessing) – 계속

C. 오후 6 ~ 8시 날씨 처리 Issue

날씨

예) 2019-02-03 날씨 데이터 / **일시 – 기온 – 강수량 – 풍속 – 적설**

```
808,108,서울,2019-02-03 17:00,4.1,0.0,2.5,0.0
809,108,서울,2019-02-03 18:00,3.9,1.0,2.1,0.0
810,108,서울,2019-02-03 19:00,3.8,0.0,2.5,0.0
811,108,서울,2019-02-03 20:00,3.7,0.0,2.8,0.0
```

[상황1] 기온이 6-7시는 3.9도, 7-8시는 3.8도

-> 6-8시 기온은 어떻게 표시?

[상황2] 강수량이 6-7시는 1.0mm, 7-8시는 0.0mm

-> 6-8시 강수량은 어떻게 표시?

2개 시간대에서 다르게 나타나는 날씨를 어떻게 처리해줄 것인가.

기온
풍속 > **평균(mean)**

강수량
적설 > **누적(sum)**

2개 시간대 → 타겟 시간대

1) 데이터 전처리 (Data Preprocessing) – 계속

D. 상권수, 사업체수 결측치 처리 Issue

상권수 사업체수

‘중구 정동’ 등 데이터가 없는 경우가 있다.

15	중구소공동	3849
16	중구회현동	8566
17	중구명동	6965
18	중구필동	2886
19	중구장충동	819
20	중구광희동	8472
21	중구을지로동	8812
22	중구황학동	1947
23	중구숭림동	1016
24	중구신당동	1193
25	중구다산동	1424
26	중구약수동	833
27	중구청구동	881
28	중구동화동	1005

‘중구’ 상권수 데이터

CASE 1 상권수

해당 동에 상권 없다고 판단되는 경우

→ 결측치 0으로 대체

CASE 2 상권수 사업체수

상권은 있지만 데이터 없는 경우

→ 소속 행정구의 median으로 대체

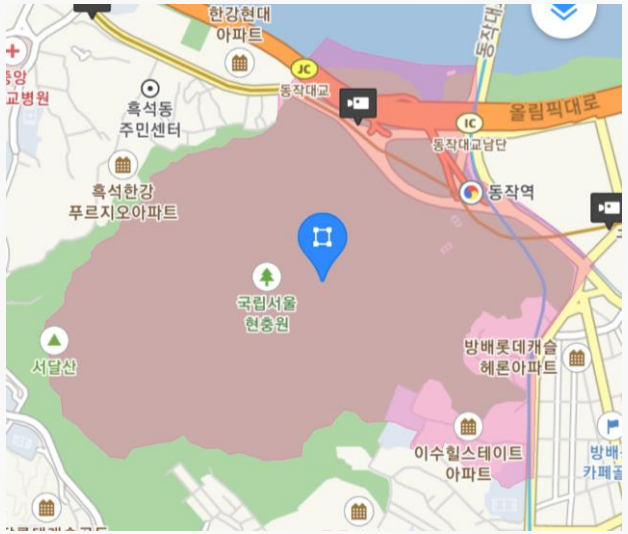
Cf) 상권 – 골목상권, 발달상권, 전통시장상권, 관광특구상권
- **골목상권** : 점포 밀집도가 높은 상권 (음식점업, 소매업, 서비스업 영위)
- **발달상권** : 도보이동 가능 범위 내의 상가업소밀집지역 (도매, 소매, 음식, 숙박, 생활서비스, 관광여가오락 등)
- **전통시장상권** : 오랜 기간에 걸쳐 일정 지역에서 자연발생적으로 형성된 상설시장이나 정기시장
- **관광특구상권** : 관광활동이 주로 이루어지는 지역적 공간 내 입지한 상권

1) 데이터 전처리 (Data Preprocessing) – 계속

D. 상권수, 사업체수 결측치 처리 Issue – 계속 상권수 사업체수

CASE 1 상권수

– 상권 없다고 판단 – 결측치 0으로 대체



동작구 동작동 카카오맵 사진

CASE 2 상권수 사업체수

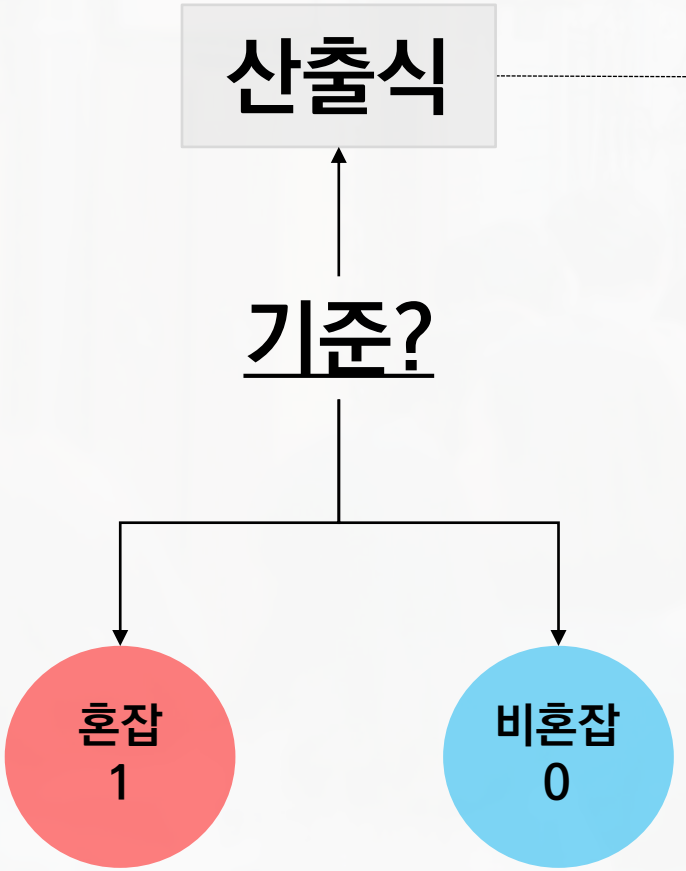
– 소속 행정구의 median*으로 대체

*극단값이 있어서 mean 아니라 median 채택

	날짜	호선	역번호	역명	요일	인원	기온	강수량	풍속	적설	연도	lat	lng	동	관광지수	사업체수
158106	2019-01-01	2호선	215.0	잠실나루	1.0	824.0	-2.95	0.0	2.90	0.0	2019	37.520669	127.103839	송파구신천동	0.0	4665.5
158471	2020-01-01	2호선	215.0	잠실나루	2.0	720.0	0.10	0.0	0.55	0.0	2020	37.520669	127.103839	송파구신천동	0.0	1089.0
158837	2019-01-01	2호선	227.0	낙성대	1.0	2589.0	-2.95	0.0	2.90	0.0	2019	37.477117	126.963418	관악구봉천동	1.0	1687.0
159202	2020-01-01	2호선	227.0	낙성대	2.0	2425.0	0.10	0.0	0.55	0.0	2020	37.477117	126.963418	관악구봉천동	1.0	880.0
159568	2019-01-01	2호선	228.0	서울대입구	1.0	4390.0	-2.95	0.0	2.90	0.0	2019	37.481210	126.952712	관악구봉천동	2.0	1687.0
159933	2020-01-01	2호선	228.0	서울대입구	2.0	4212.0	0.10	0.0	0.55	0.0	2020	37.481210	126.952712	관악구봉천동	2.0	880.0
160299	2019-01-01	2호선	229.0	봉천	1.0	1673.0	-2.95	0.0	2.90	0.0	2019	37.482476	126.941574	관악구봉천동	0.0	1687.0
160664	2020-01-01	2호선	229.0	봉천	2.0	1812.0	0.10	0.0	0.55	0.0	2020	37.482476	126.941574	관악구봉천동	0.0	880.0
161030	2019-01-01	2호선	239.0	홍대입구	1.0	9830.0	-2.95	0.0	2.90	0.0	2019	37.557527	126.924467	마포구동교동	0.0	2576.5

사업체수 대체 결과

2) 학습 데이터 혼잡 여부 라벨링 (Labeling)



‘혼잡도’ 산출식 관련 문헌 및 사례 조사

✓ **서울교통공사 혼잡도 기준**

– 신형전동차 혼잡도는 차체 하중을 연산하여 표시하며, 정원의 130% 이상을 혼잡으로 정의

✓ **대도시권광역교통위원회, 도시철도의 건설과 지원에 관한 기준**

– 혼잡도가 150%를 넘으면 매우 혼잡하다고 판단하여 노선 증량 및 증편 요구

✓ **김진수(2016), 빅데이터 분석을 이용한 지하철 혼잡도 예측 및 추천시스템**

– 차내 혼잡도 = Congestion(탑승인원) / Capacity(용량)

– 인천교통공사도 해당 지표 활용

✓ **김해 외(2018), 유동인구 빅데이터 기반 고속도로 휴게소 혼잡지표 개발 연구**

– 대안1 : 혼잡지표 = 유동인구(인) / 건물면적(m^2)

– 대안2 : 혼잡지표 = 유동인구(인) / 수용인원(m^2)

✓ **김진아 외(2020), 빅데이터 분석을 활용한 공항 혼잡도 분석 – 김포공항 사례를 중심으로**

– 공항 내 시설별 적정 인원을 도출 → 적정 인원의 130% 이상을 ‘혼잡’으로 정의

2) 학습 데이터 혼잡 여부 라벨링 (Labeling) – 계속

① 산출식 논의 – 3안 채택

1안*) 현재 인원 / 용량
2안**) 현재 인원 / 면적
3안) 현재 인원 / 적정 인원

② 산출식 세팅

해당 역의 하차 인원
전체 역의 평균 하차 인원

= 전체 역의 적정 인원 대비 얼마나 하차하는지?

③ 혼잡 여부 라벨링

산출식 〉= 1.3
- T : 혼잡 (1)
- F : 비혼잡 (0)

*혼잡과 비혼잡 구분 기준 130%로 차용

* 1안 관련 : 역 부근의 최대 인원(용량) 계산 불가
** 2안 관련 : 역의 면적으로 계산하는 것이 논리 타당성X

④ 모델 문제 발견

– 산출식을 잘못 세팅
– '전체 역의 평균 하차인원'이 문제
– 지나치게 치우친 역들이 있음
→ 혼잡여부가 결정되는 꼴

〉 기본 모델로 시범 테스트 진행

```
rf = RandomForestClassifier()
rf.fit(X_train, y_train)
pred = rf.predict(X_test)
```

*Accuracy : 0.94, F1 Score : 0.87

⑤ 산출식 다시 세팅

해당 역의 하차 인원
해당 역의 하차 인원 중간값

= 해당 역의 적정 인원 대비 얼마나 하차하는지?
* 이상치(Outlier) 고려한 중간값(Median) 처리

2) 학습 데이터 혼잡 여부 라벨링 (Labeling) – 계속

최종 데이터

	역명	요일	인원	기온	강수량	풍속	적설	연도	lat	lng	동	관광지수	사업체수	인구수	상권수	혼잡	new_비율
0	청량리	1.0	1576.0	-2.95	맑음	2.90	맑음	2019	37.581381	127.048958	동대문구전농동	0.0	3182.0	51065.0	9.0	0	0.368483
1	청량리	2.0	4327.0	-2.95	맑음	2.10	맑음	2019	37.581381	127.048958	동대문구전농동	0.0	3182.0	51065.0	9.0	0	1.011690
2	청량리	3.0	4304.0	-0.65	맑음	1.80	맑음	2019	37.581381	127.048958	동대문구전농동	0.0	3182.0	51065.0	9.0	0	1.006313
3	청량리	4.0	4711.0	1.90	맑음	1.25	맑음	2019	37.581381	127.048958	동대문구전농동	0.0	3182.0	51065.0	9.0	0	1.101473
4	청량리	5.0	2734.0	-1.80	맑음	1.55	맑음	2019	37.581381	127.048958	동대문구전농동	0.0	3182.0	51065.0	9.0	0	0.639233
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
181494	서울역	6.0	489.0	7.55	맑음	1.60	맑음	2020	37.555946	126.972317	용산구동자동	2.0	1212.0	13622.0	4.0	0	0.341242
181495	서울역	0.0	1063.0	7.70	맑음	1.60	맑음	2020	37.555946	126.972317	용산구동자동	2.0	1212.0	13622.0	4.0	0	0.741800
181496	서울역	1.0	1150.0	-0.50	맑음	3.70	맑음	2020	37.555946	126.972317	용산구동자동	2.0	1212.0	13622.0	4.0	0	0.802512
181497	서울역	2.0	1311.0	-10.95	맑음	4.40	맑음	2020	37.555946	126.972317	용산구동자동	2.0	1212.0	13622.0	4.0	0	0.914864
181498	서울역	3.0	1197.0	-6.90	맑음	2.45	맑음	2020	37.555946	126.972317	용산구동자동	2.0	1212.0	13622.0	4.0	0	0.835311

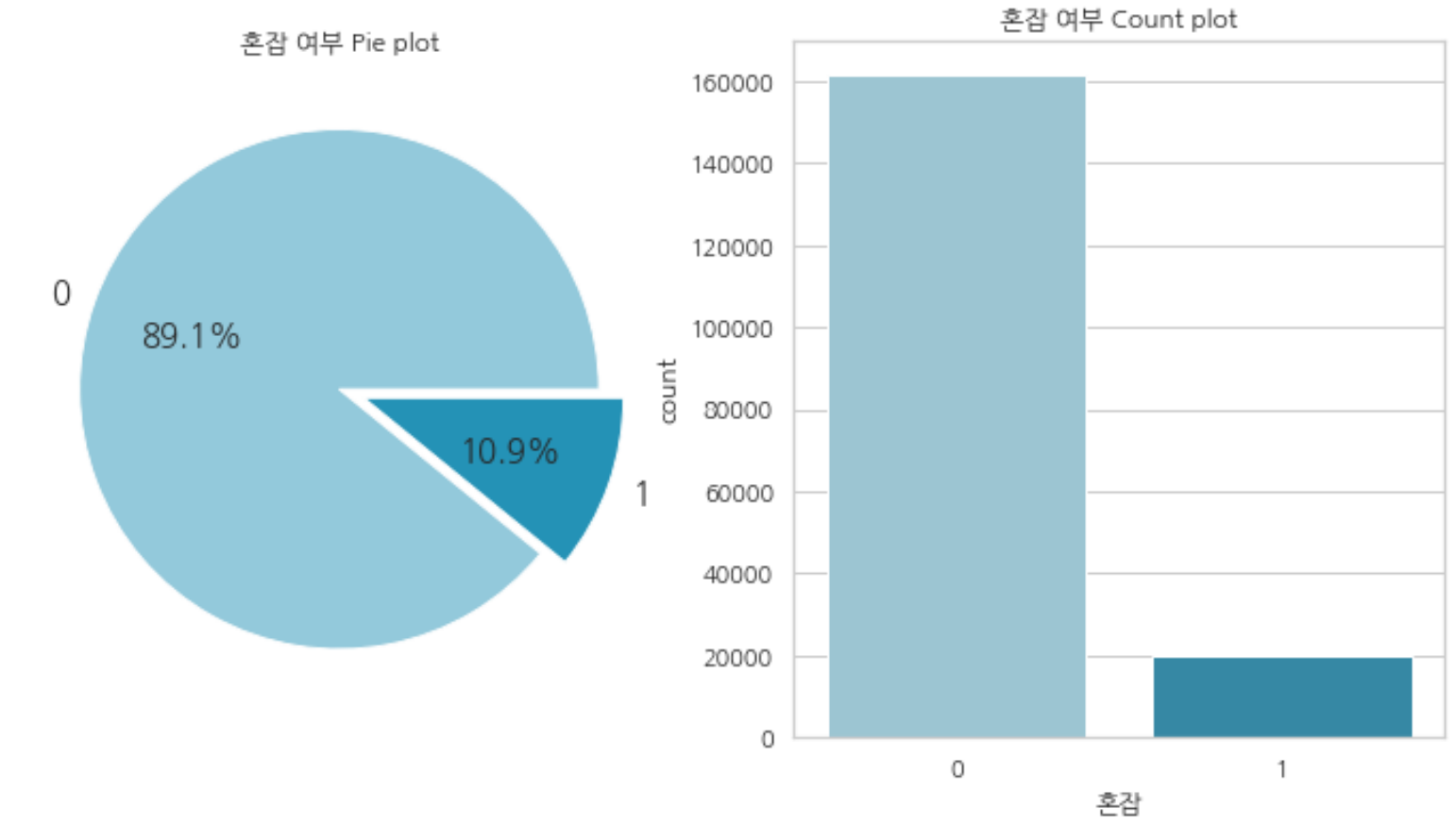
181499 rows × 17 columns

총 181,499개 데이터

- Target(Y) : 혼잡 (0 : 비혼잡, 1 : 혼잡)
- Feature(X) :
  - ✓ 범주형(Categorical) : 역명, 요일, 강수량*, 적설*
  - ✓ 연속형(Continuous) : 관광지수, 사업체수, 인구수, 상권수, 기온, 풍속

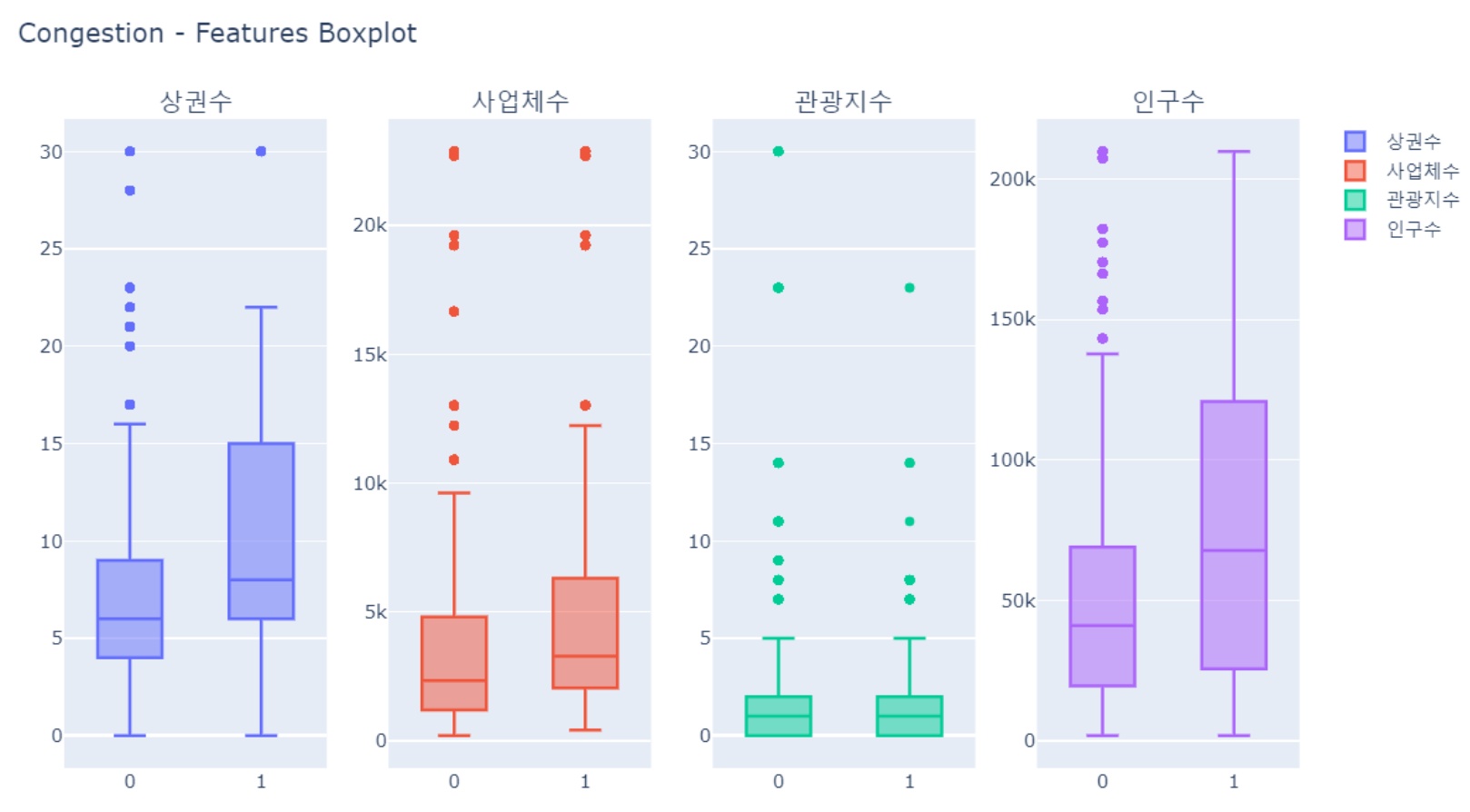
* 강수량과 적설은 구간별로 범주화함 – 강수량 : 맑음, 비 매우 조금, 비 조금 비 다소, 비 다소 많음 / 적설 : 맑음, 눈 조금, 눈 다소, 눈 다소 많음 (기준:웨더아이 참고)

3) 데이터 시각화 * Target(혼잡 여부)별 수



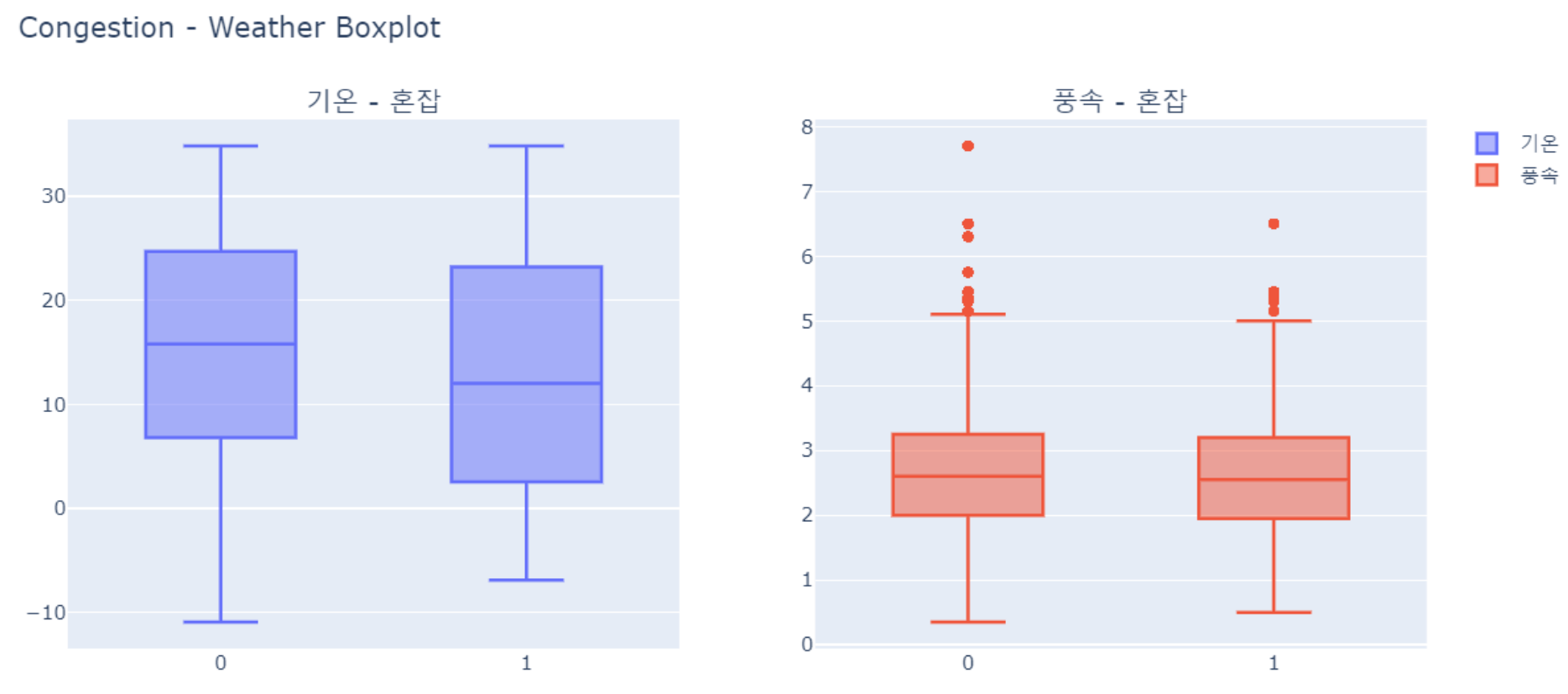
- 혼잡 여부(0, 1)의 수 차이가 크다. → **라벨 불균형**
- 혼잡(1)이 10.9%, 비혼잡(0)이 89.1%

3) 데이터 시각화 * 혼잡여부별 상권수, 사업체수, 관광지수, 인구수 분포



- 상권수, 사업체수, 인구수는 혼잡한 역 인근에 더 크게 나타나고 있다.
- 관광지수는 혼잡과 비혼잡 사이에 차이가 크게 보이지 않는다.

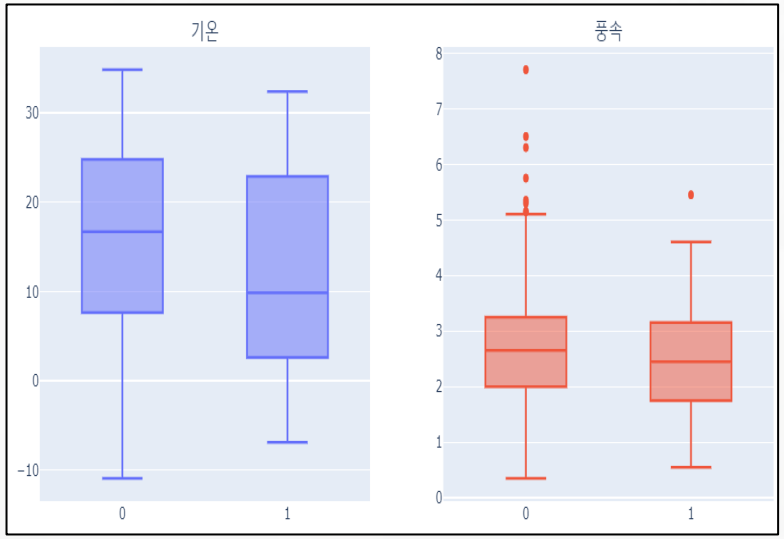
3) 데이터 시각화 * 혼잡여부별 날씨(기온과 풍속) 분포



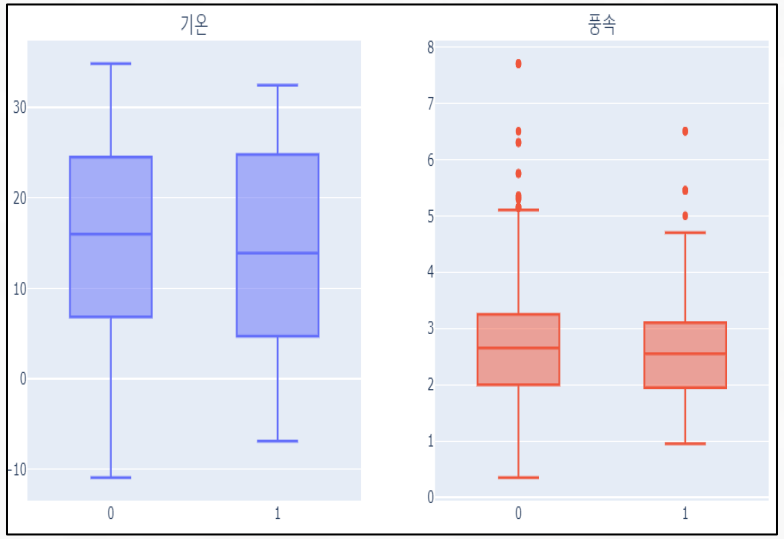
- 기온은 다소 낮을수록 혼잡한 경우가 많다.
- 풍속이 매우 큰 경우에는 오히려 혼잡하지 않게 나타난다.

3) 데이터 시각화 * 혼잡여부별 날씨(기온과 풍속) 분포 – 3개 혼잡 지하철역

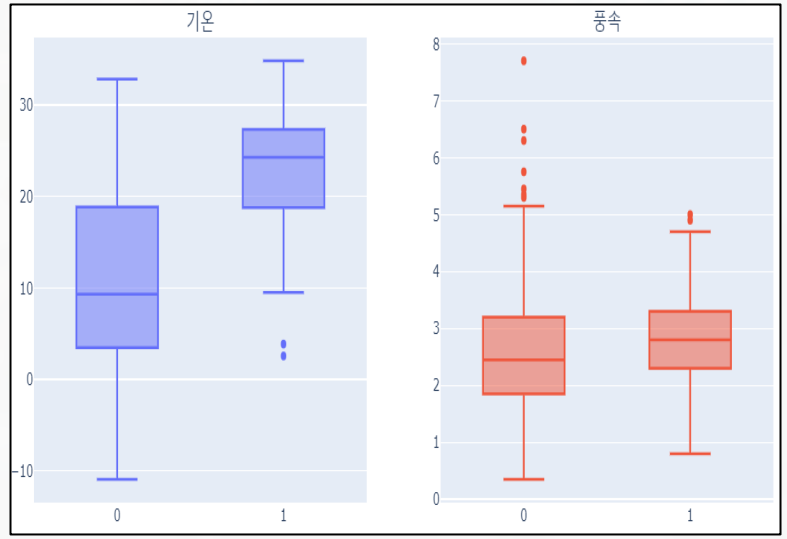
〈서울역〉



〈종로3가역〉



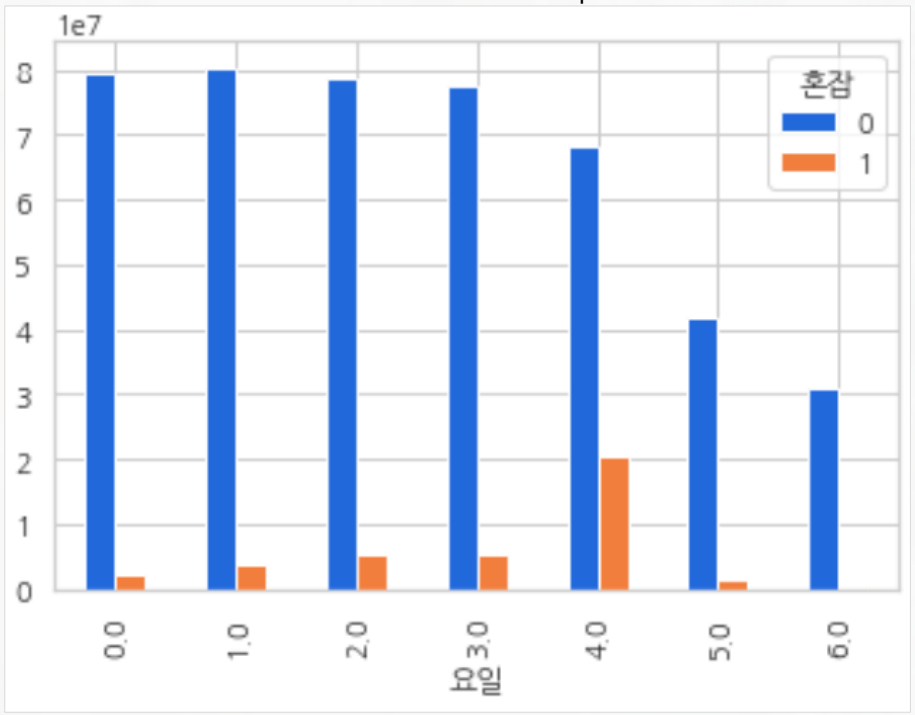
〈여의나루역〉



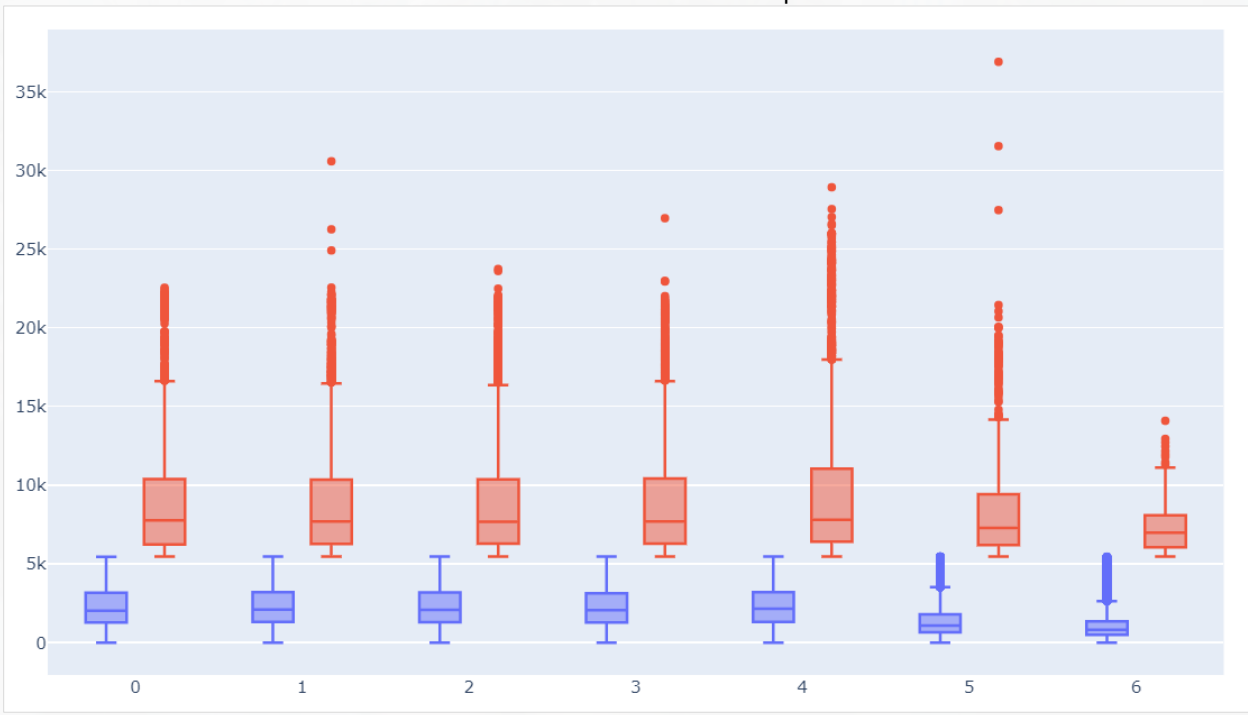
– 서울역과 종로3가역은 유사한 분포를 보인다 : 기온, 풍속이 낮을수록 혼잡한 것으로 나타난다.
– 반면, 여의나루역은 기온이 상대적으로 높아지면 혼잡해진다 : '한강공원'이라서 기온이 높아지면 혼잡한 경향
– 3개역만 놓고 보면, 지하철역 인근 지역의 특성이 반영될 가능성이 있다.

3) 데이터 시각화 * 요일별 – 하차 인원 : 혼잡(1) / 비혼잡(0) 구분

요일별 하차인원 Barplot

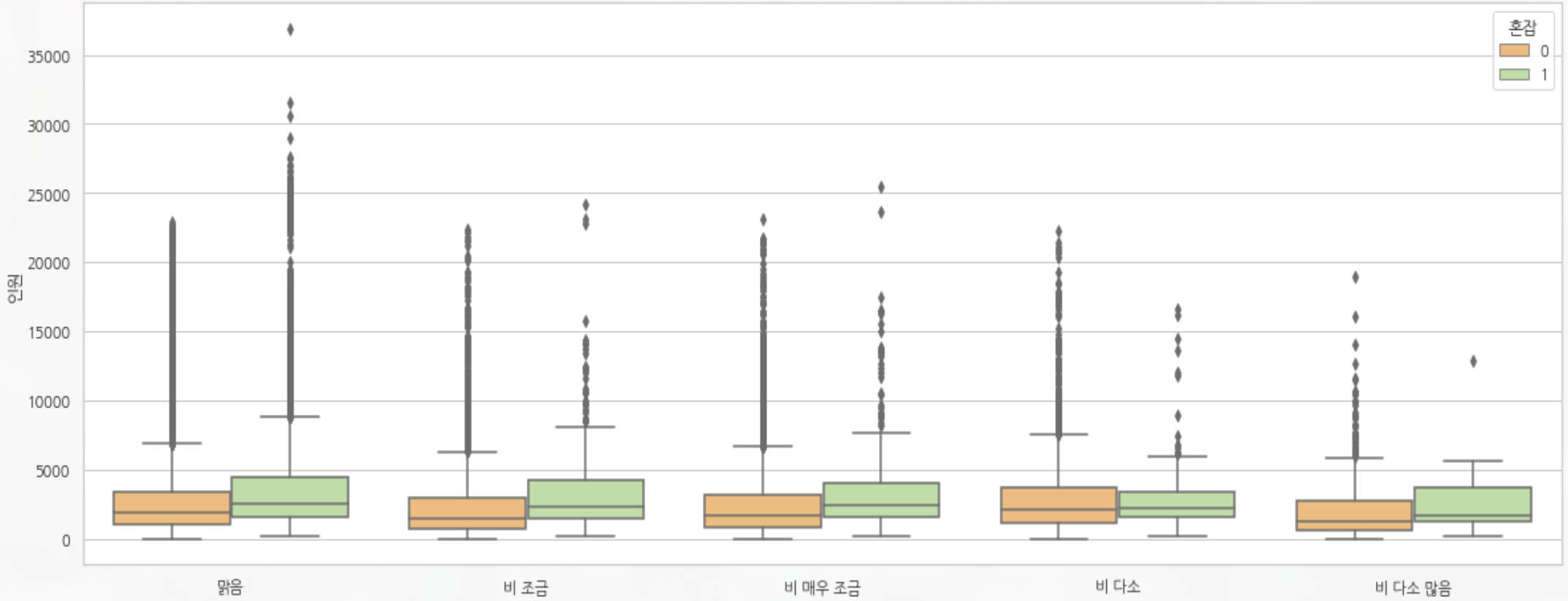


요일별 하차인원 Boxplot



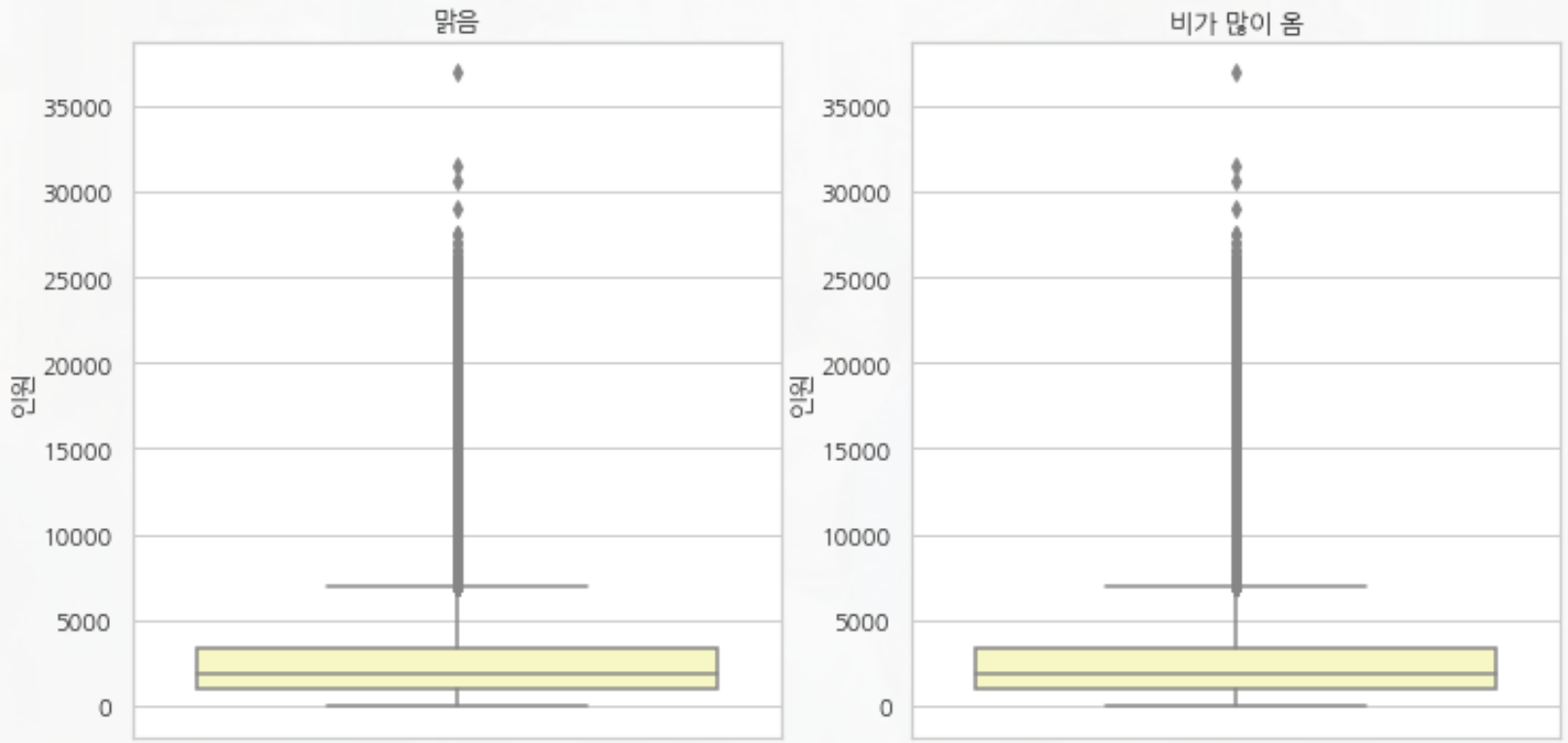
- 주말(토/일)에 하차 인원 수준이 낮게 나타난다 : 특히 비혼잡의 경우에 하차인원이 다른 요일에 비해 현저히 낮다.
- 월 ~ 금은 혼잡, 비혼잡 모두 인원 수 거의 변화 없다. : 금요일 혼잡 지하철역이 미세하게 인원이 높게 나타나고 있다.
- 토요일 혼잡 역의 경우, Outlier가 매우 높게 나타난다. : '잠실, 서초, 여의나루'
- 잠실은 연휴(5/4, 5, 6), 서초는 집회, 여의나루는 불꽃축제의 영향을 받은 것으로 추정

3) 데이터 시각화 * 혼잡여부별 강수량별 하차 인원 분포



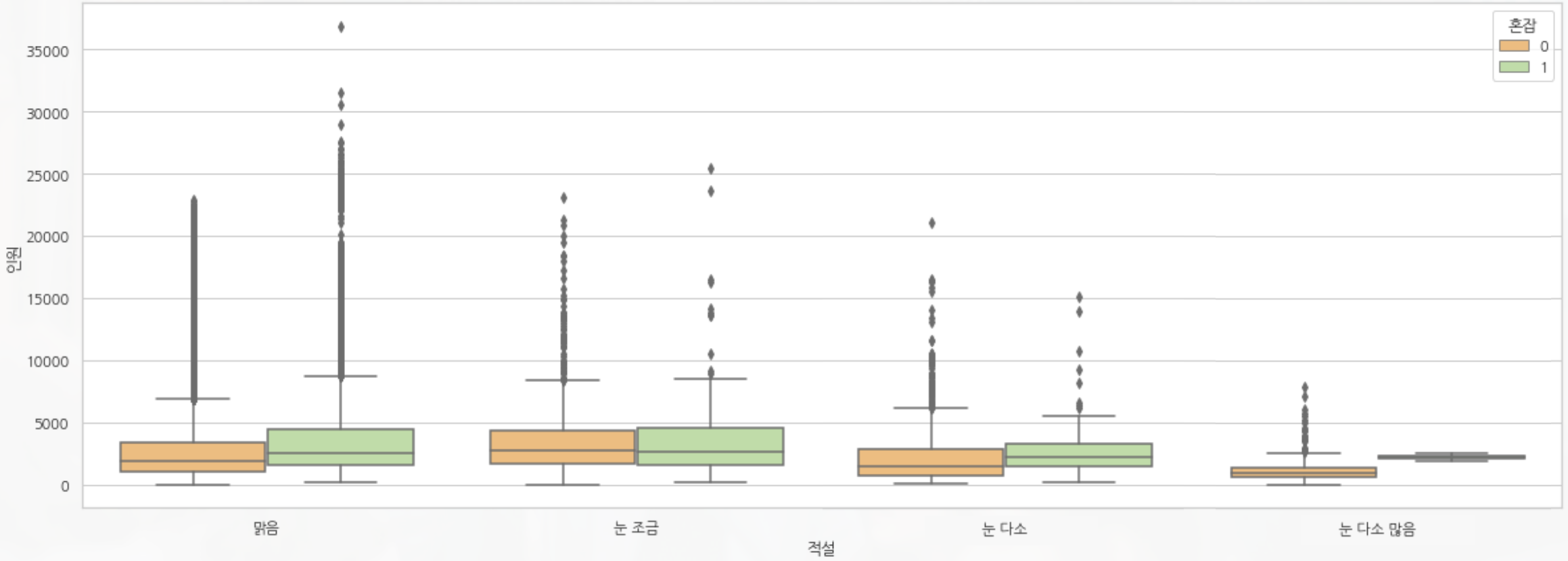
- 비의 유무에 따라서 하차인원이 매우 근소하게 달라진다.
- 강수의 정도가 역 부근의 혼잡도에 영향을 미치지 않는 것으로 보인다.

3) 데이터 시각화 * 강수량별 하차 인원 분포 – 맑음 vs 비 많이 옴 (비가 다소, 비 다소 많음)



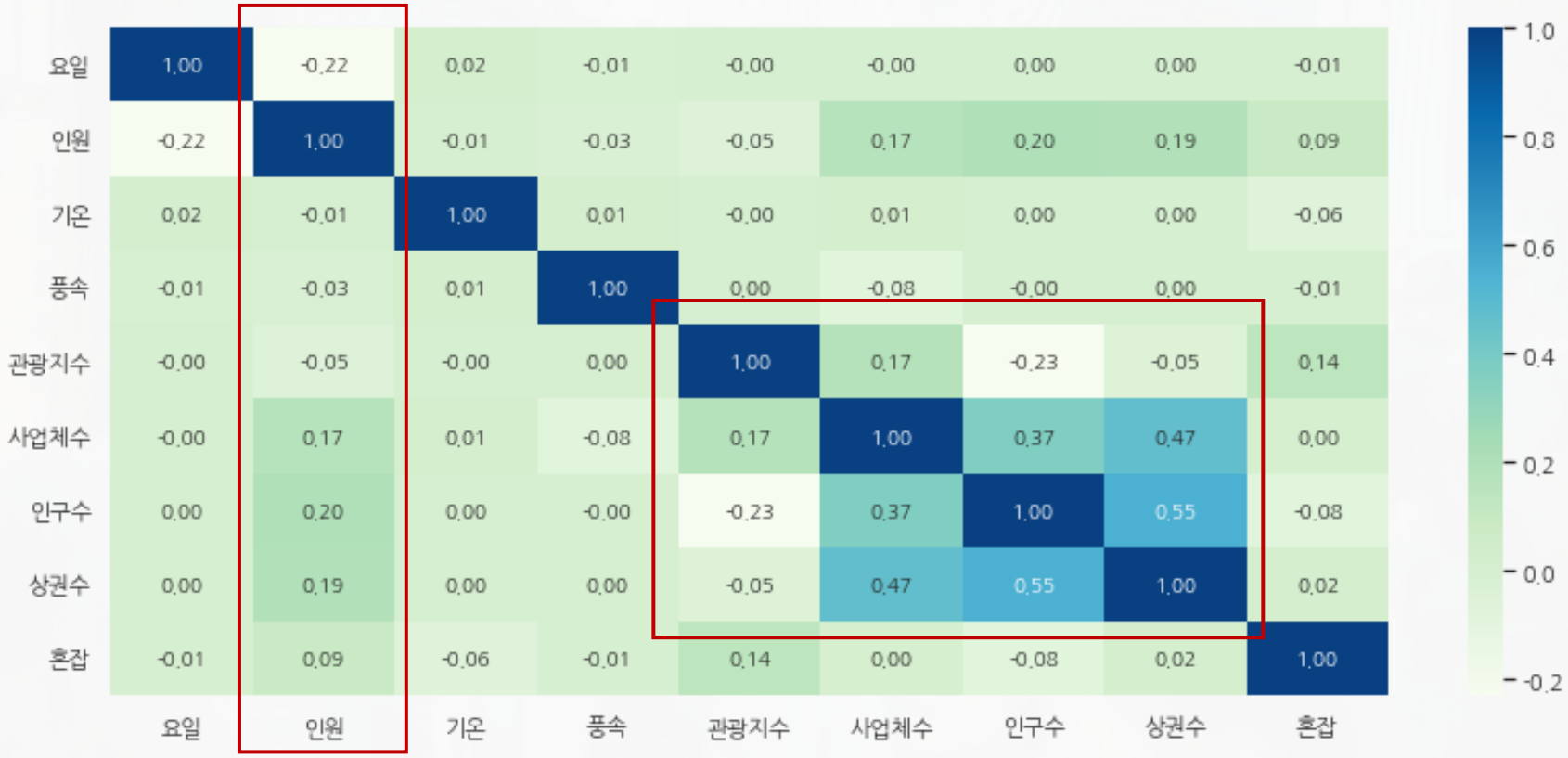
- 거의 차이가 없다.

3) 데이터 시각화 * 혼잡여부별 적설량별 하차 인원 분포



- 눈이 오면 하차 인원이 적어지는 경향을 보인다.
- 적설량이 증가할수록 더욱 경향이 짙어진다.

3) 데이터 시각화 * 히트맵 (Heatmap)



- '하차인원'과 연관된 특성 : (big) 사업체수, 인구수, 상권수, (small) 기온, 풍속, 관광지수
- 상관관계 : 상권수 – 인구수 / 사업체수 – 상권수 / 사업체수 – 인구수 / 관광지수–사업체수

3) 데이터 시각화 * 한번도 혼잡하지 않았던 역 vs 혼잡/비혼잡 공존했던 역

0

2년간 항상 혼잡했던 역
(always_conges)

71

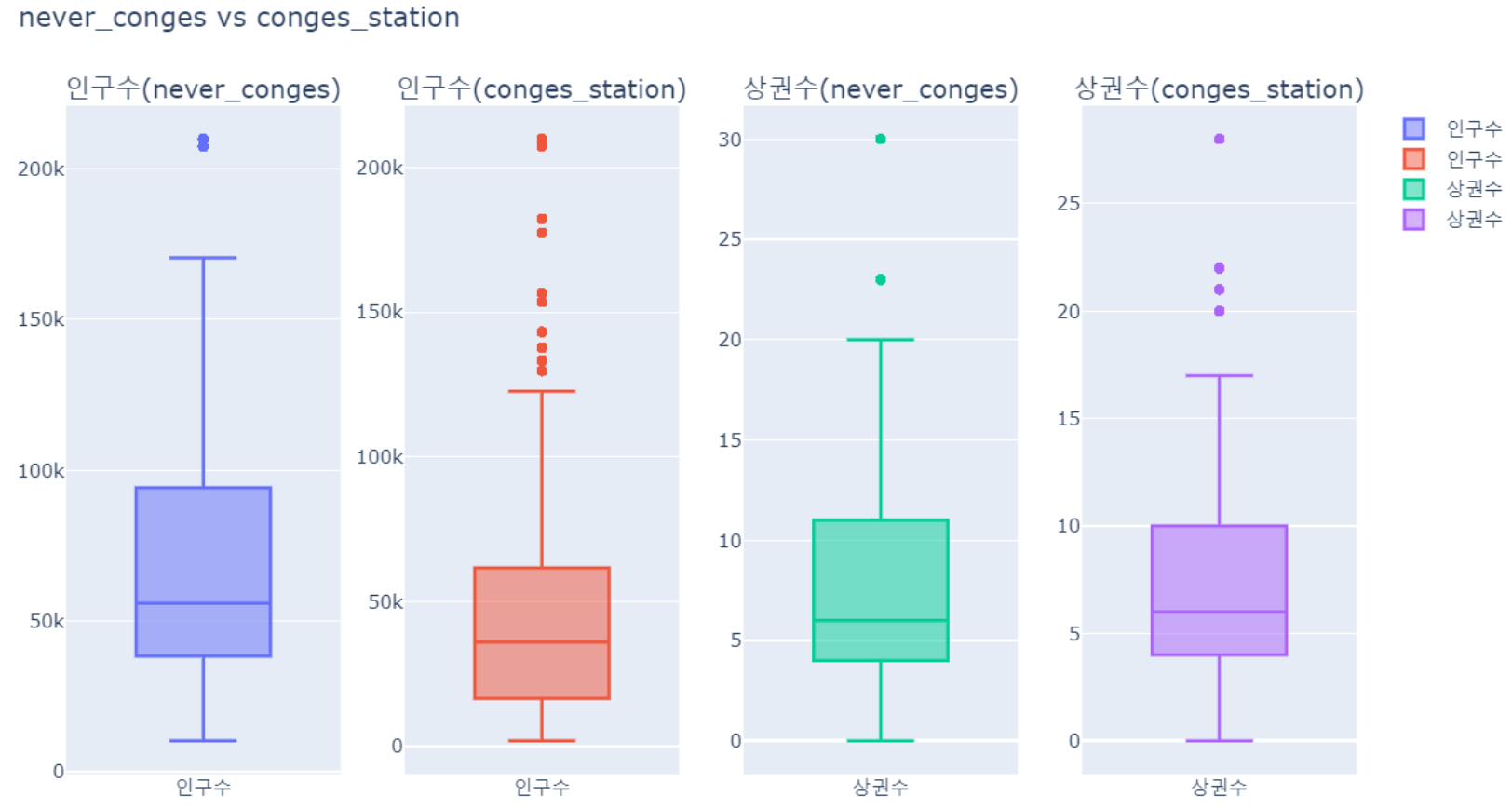
한번도 혼잡하지 않은 역
(never_conges)

146

혼잡/비혼잡 공존한 역
(conges_station)

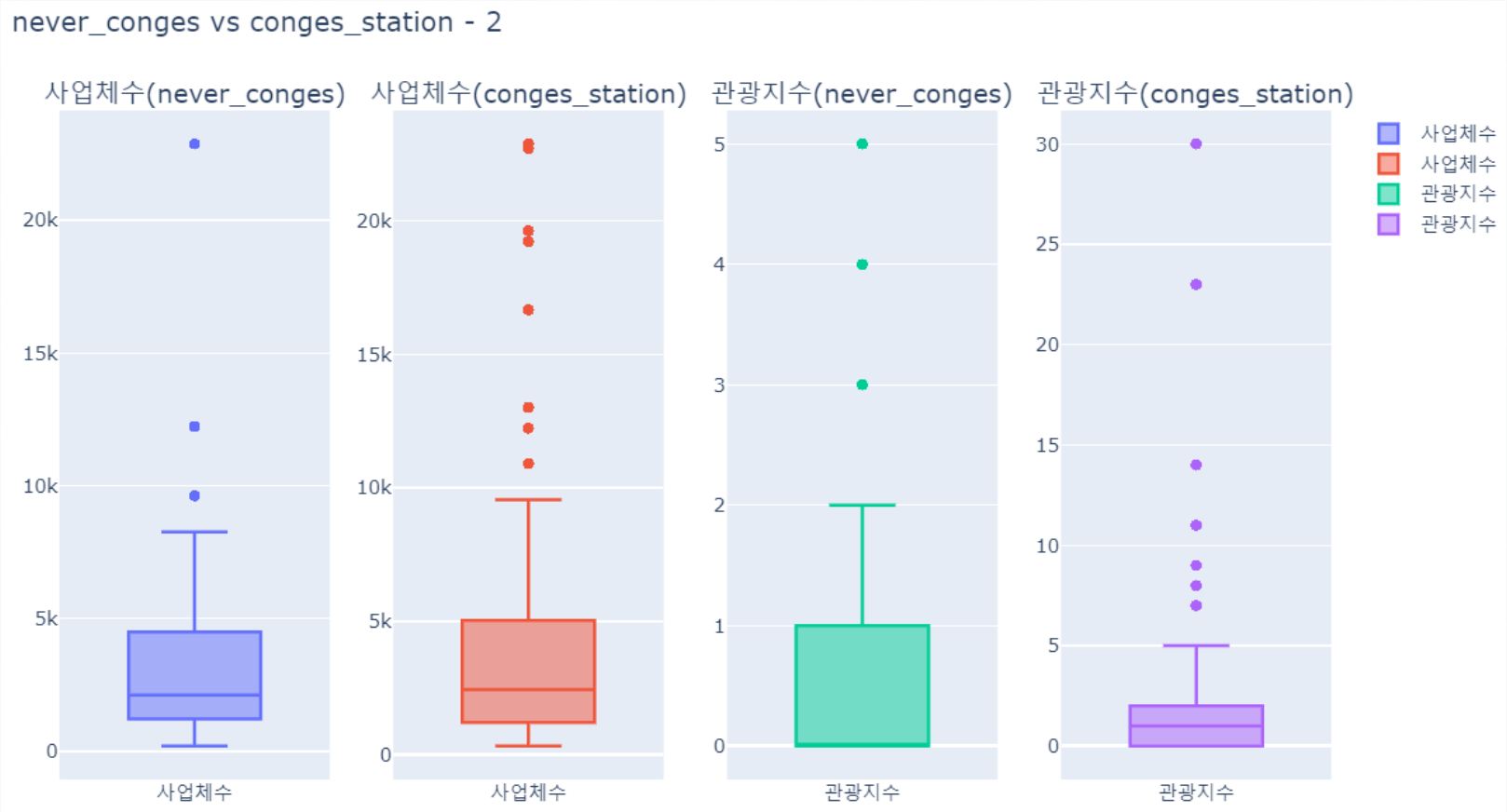
특성별 비교 필요

3) 데이터 시각화 * 한번도 혼잡하지 않았던 역 vs 혼잡/비혼잡 공존했던 역 – 인구수, 상권수



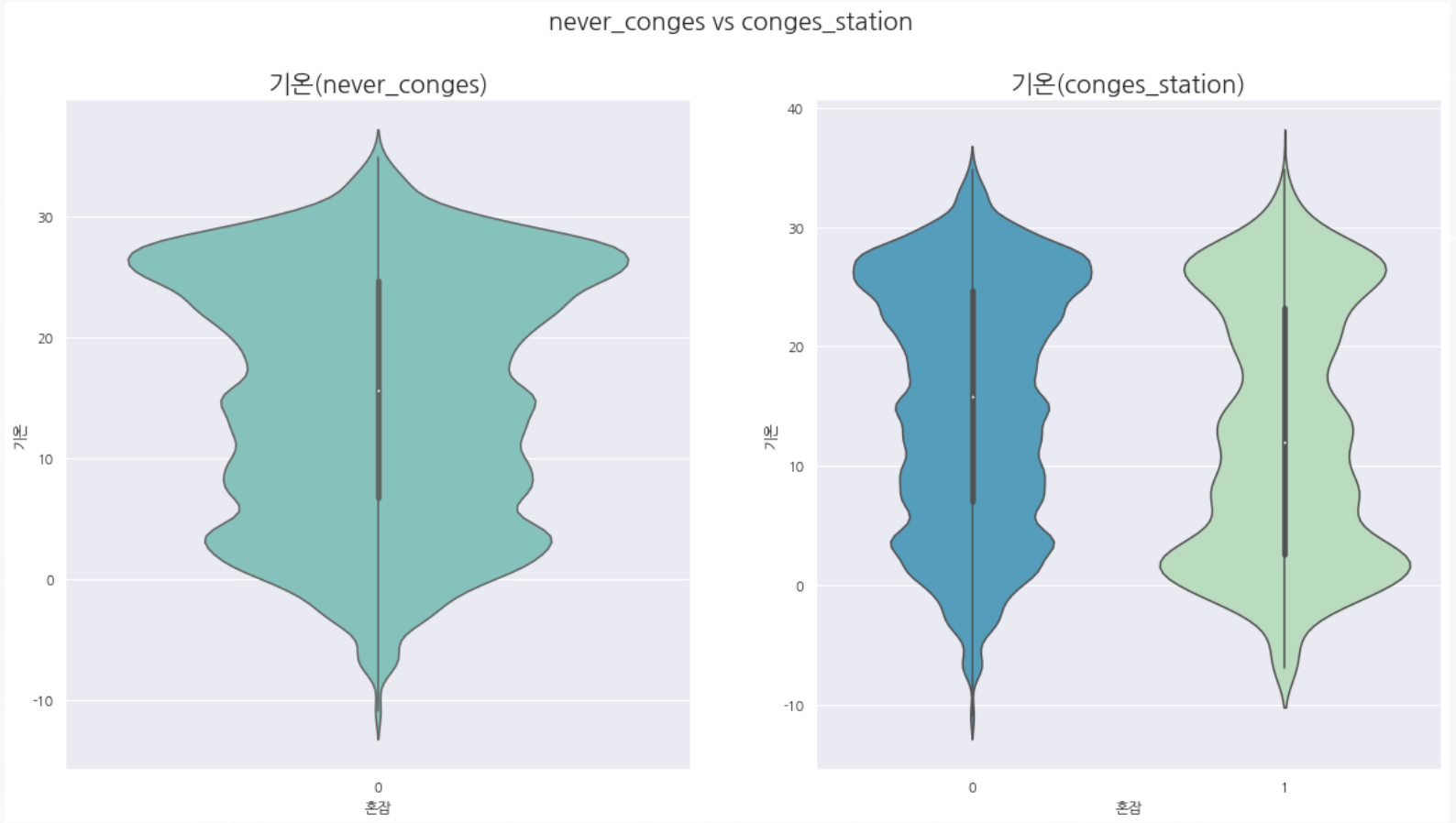
- 대체적으로 한번도 혼잡하지 않았던 역의 인구수와 상권수가 높게 나타난다.
- 인구수가 많지 않아도 번화가나 교통 관련 역이 많이 포함되어 있다면, 혼잡해도 인구수가 적게 나타나는 결과가 나올 수 있다.

3) 데이터 시각화 * 한번도 혼잡하지 않았던 역 vs 혼잡/비혼잡 공존했던 역 – 사업체수, 관광지수



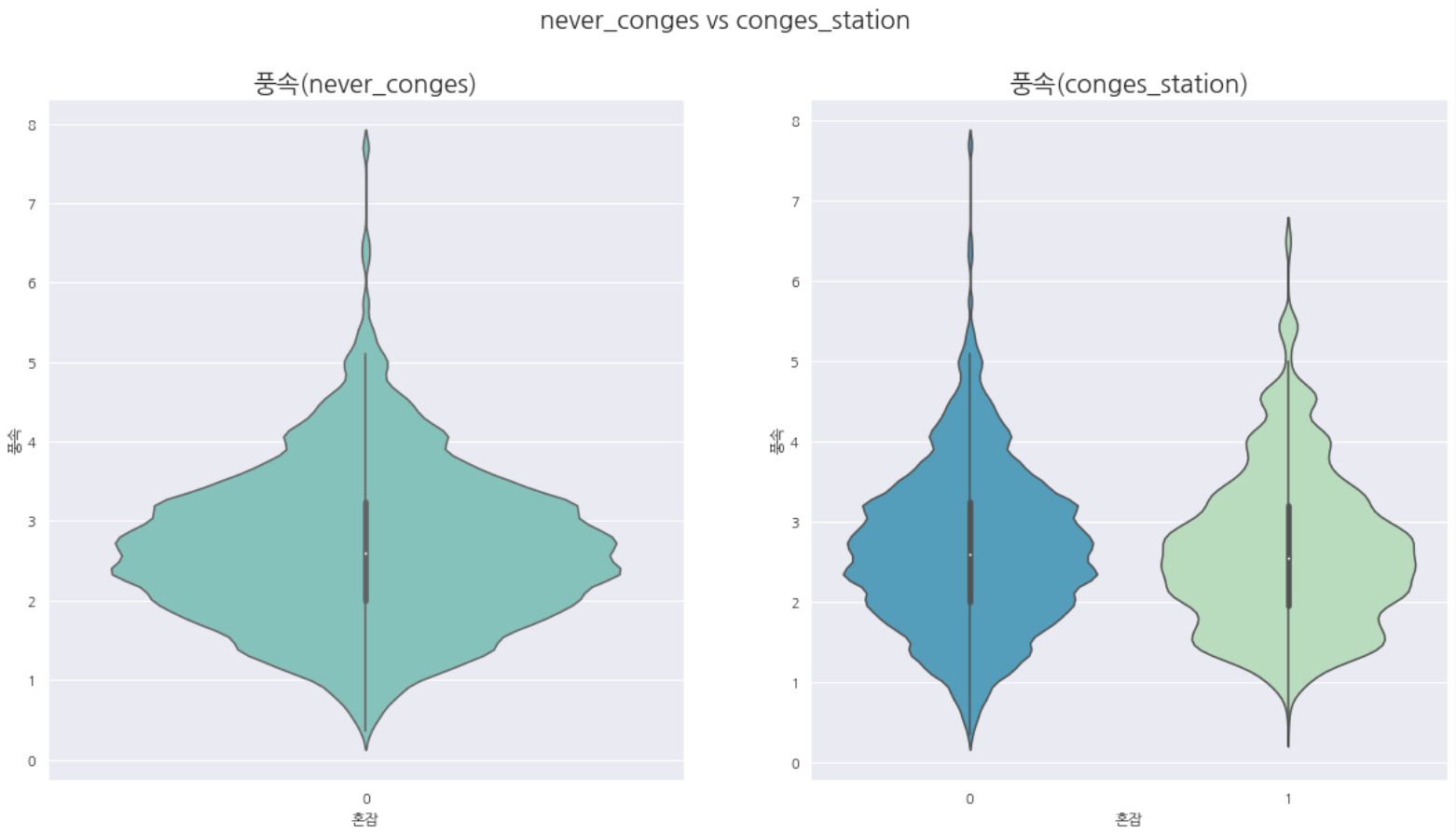
- 출퇴근의 영향으로 사업체수가 많을수록 혼잡하며, 관광지수는 혼잡에 영향을 주지 않는 듯하다.
- 유동인구의 대부분은 출퇴근의 목적일 가능성이 많다는 것을 다시 한 번 알 수 있다.

3) 데이터 시각화 * 한번도 혼잡하지 않았던 역 vs 혼잡/비혼잡 공존했던 역 – 기온



- 혼잡한 역의 경우 기온 차이에 따라 분포 차이가 커진다.
- 매우 추운 경우(-10도 이하)에는 역이 혼잡한 경우가 없다.

3) 데이터 시각화 * 한번도 혼잡하지 않았던 역 vs 혼잡/비혼잡 공존했던 역 – 풍속



- 바람이 매우 세게(7 이상) 불면, 역이 혼잡하지 않는 것으로 나타난다.
- 풍속은 둘 사이의 차이가 크게 나타나지는 않는다.

3. 모델링

1) 모델 기본 설계 (Model Design)

Scaler	Standard Scaler
Clf	1) DecisionTreeClassifier / 2) AdaBoostClassifier / 3) GradientBoostingClassifier 4) RandomForestClassifier / 5) LogisticRegression / 6) LGBMClassifier
Target	혼잡 여부(0, 1) → Upsampling(SMOTE) *라벨 : 하차인원 / 해당 역 평균 하차인원
Feature	['요일', '기온', '강수량', '풍속', '적설', '관광지수', '사업체수', '인구수', '상권수', '역명_label']
Split	Train : Validation : Test (%) = 64 : 16 : 20

2) 모델 선택 (Model Selection)

Pipeline

Scaler → Classifier(clf)

Scaler	Classifier(clf)
StandardScaler	DecisionTreeClassifier AdaBoostClassifier GradientBoostingClassifier RandomForestClassifier LogisticRegression LGBMClassifier

- Standard Scaler와 6종의 분류(Classification) 분류기를 Pipeline으로 구축하여 Score 비교

Final Model !!

3) 특성 선택 (Feature Selection)

✓ 각 Case 별로 Feature Selection 진행해서 모델 결과를 각각 확인하였음

CASE 1	CASE 2	CASE 3	CASE 4	CASE 5
모든 특성 선택	**적설, 강수량 제외**	**관광지수, 상권수 제외**	**요일 제외**	**날씨 전체 제외**
Feature Importance 확인	중요도 낮은 Feature	일별 고정 Feature 중 중요도 낮은 Feature	주기성 Feature	일별 변화 Feature

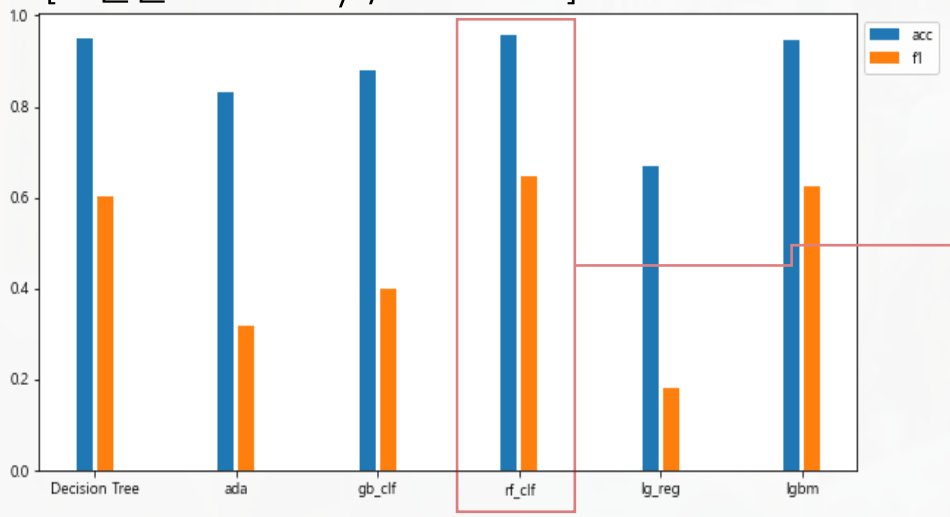
4. 프로젝트 결과

모델 성능 평가 (Score)

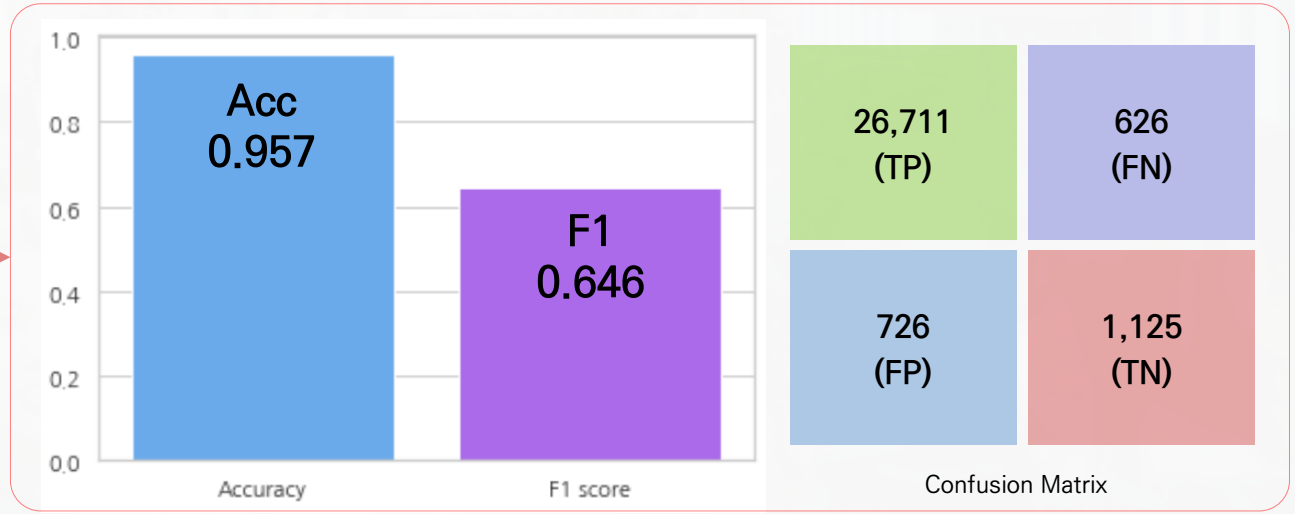
CASE 1 모든 특성 선택

*상세 Score는 Appendix에서 확인 가능

[모델별 Accuracy / F1 Score]



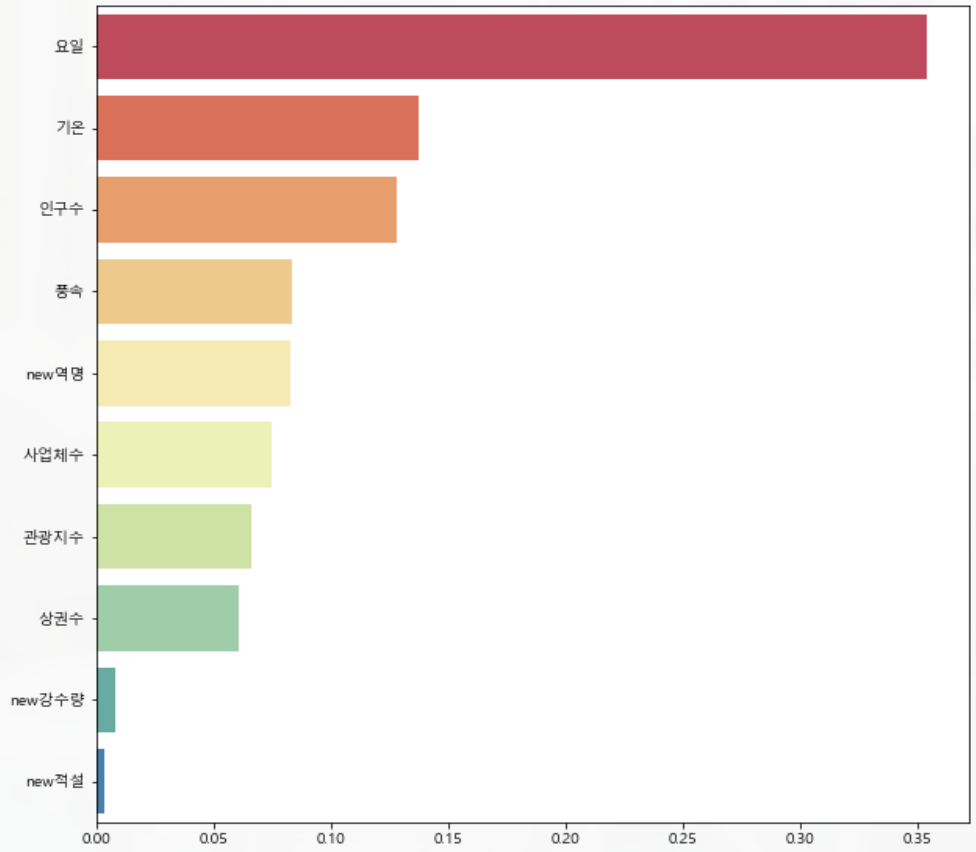
[Best Model (RandomForest)]



- RandomForest 성능이 가장 좋고, Logistic Regression 성능은 매우 좋지 않다.
- 모든 특성을 선택해 모델링하면 Accuracy 약 0.957, F1 Score 약 0.646의 성능을 보인다.

모델 성능 평가 (Score) – 계속

CASE 1 모든 특성 선택



[Best Model의 Feature Importance]

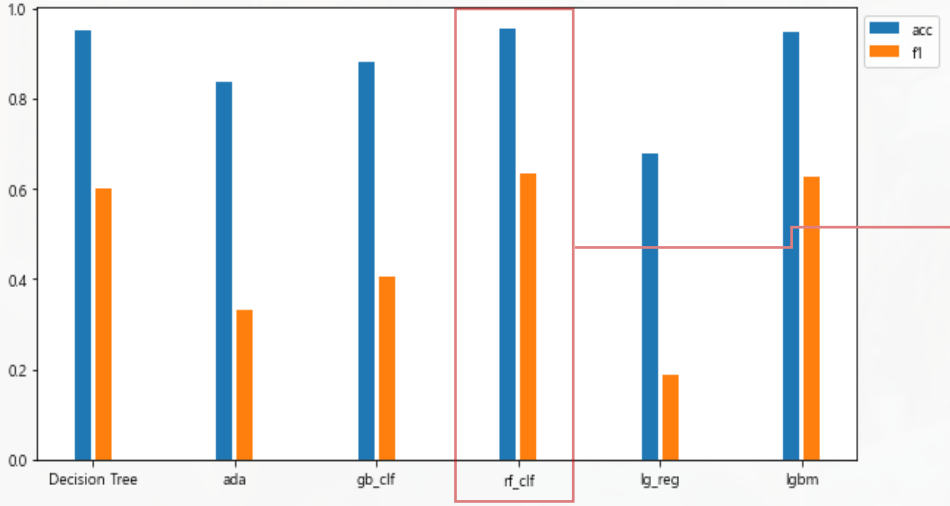
- 요일, 기온, 인구수 순으로 중요도가 높게 나타났다.
- 강수량과 적설이 중요도가 가장 낮게 나타났다.
- 풍속, 역명, 사업체수, 관광지수, 상권수는 비슷하게 반영되고 있다.

모델 성능 평가 (Score) – 계속

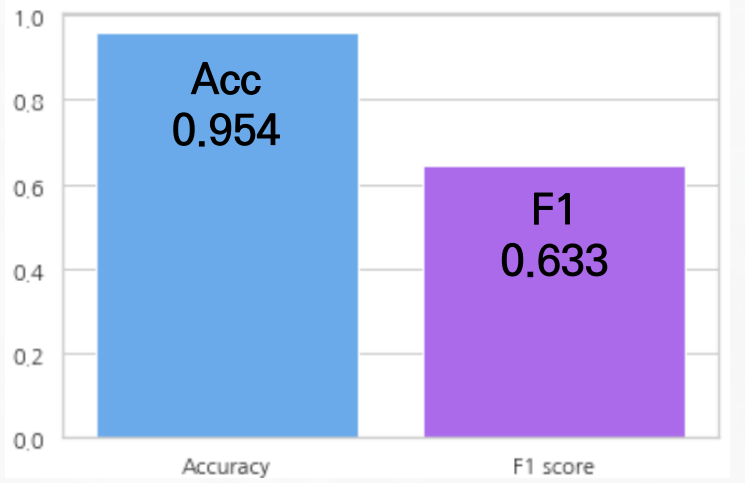
CASE 2 적설, 강수량 제외

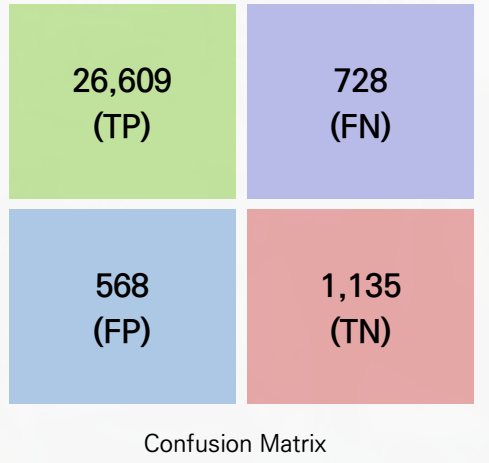
*상세 Score는 Appendix에서 확인 가능

[모델별 Accuracy / F1 Score]



[Best Model (RandomForest)]

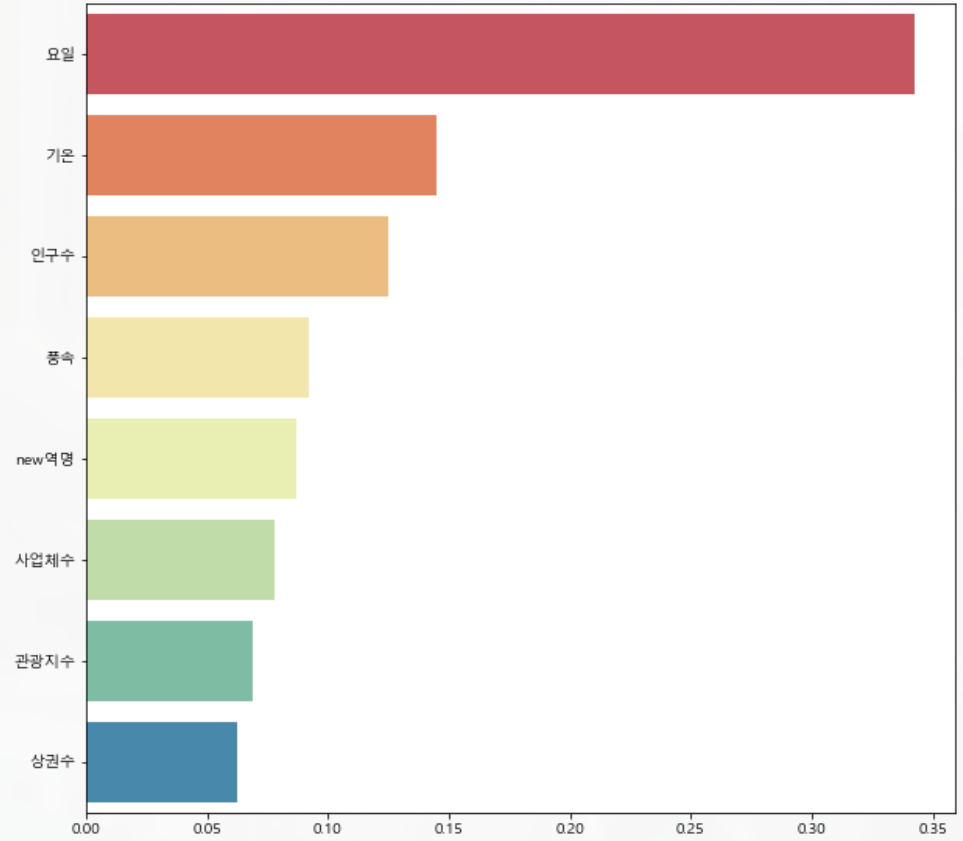




- Case1(RF기준)에서 중요도가 낮았던 적설, 강수량을 제외한 모델은 제외하지 않았을 때(모든 특성)와 비교하여 F1 Score가 약 0.013 하락했다.

모델 성능 평가 (Score) – 계속

CASE 2 적설, 강수량 제외



[Best Model의 Feature Importance]

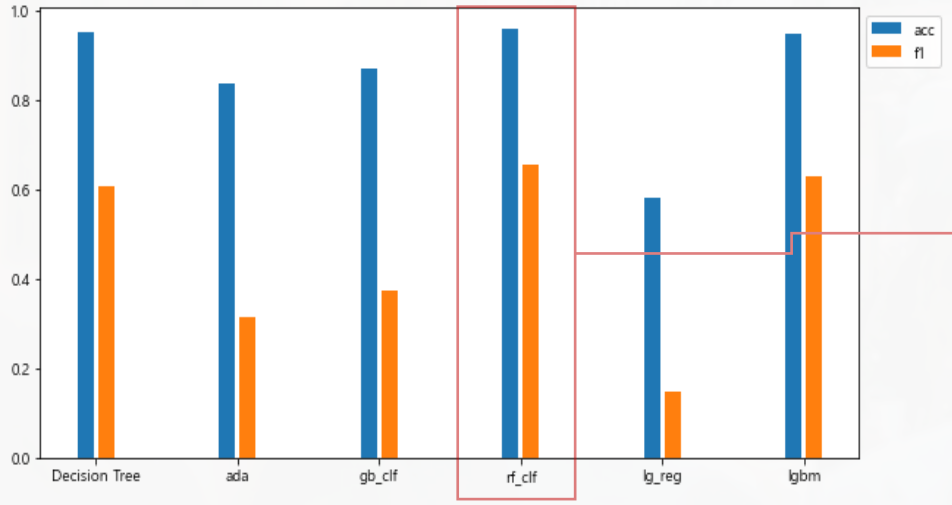
- Case 1 Best Model의 중요도와 동일한 순서로 나타났다.
- 요일, 기온, 인구수 순으로 중요도가 높게 나타났다.
- 상위 3개 Feature를 제외하고는 거의 비슷하게 반영되고 있다.

모델 성능 평가 (Score) – 계속

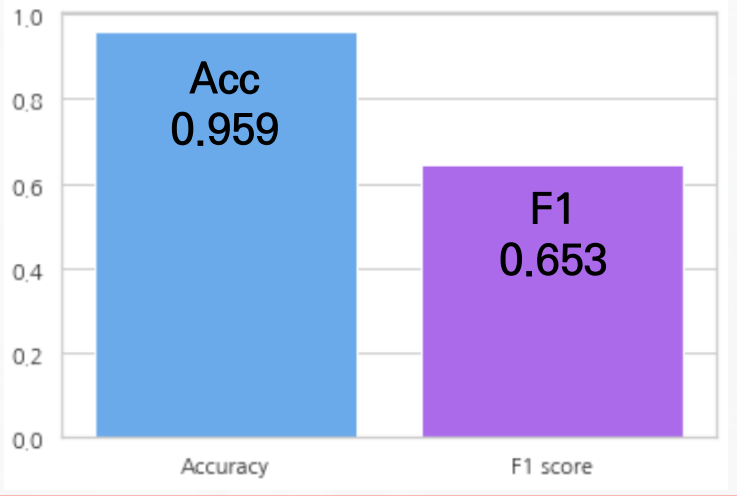
CASE 3 **관광지수, 상권수 제외**

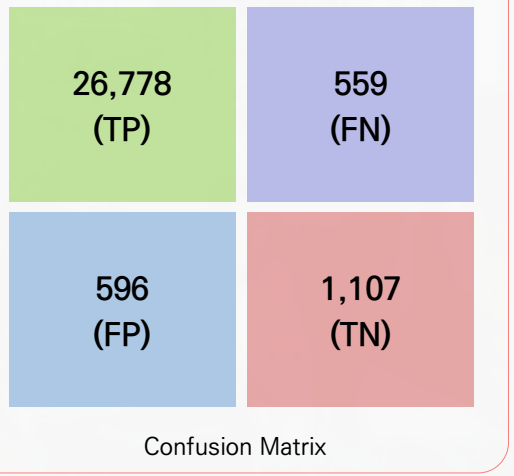
*상세 Score는 Appendix에서 확인 가능

[모델별 Accuracy / F1 Score]



[Best Model (RandomForest)]

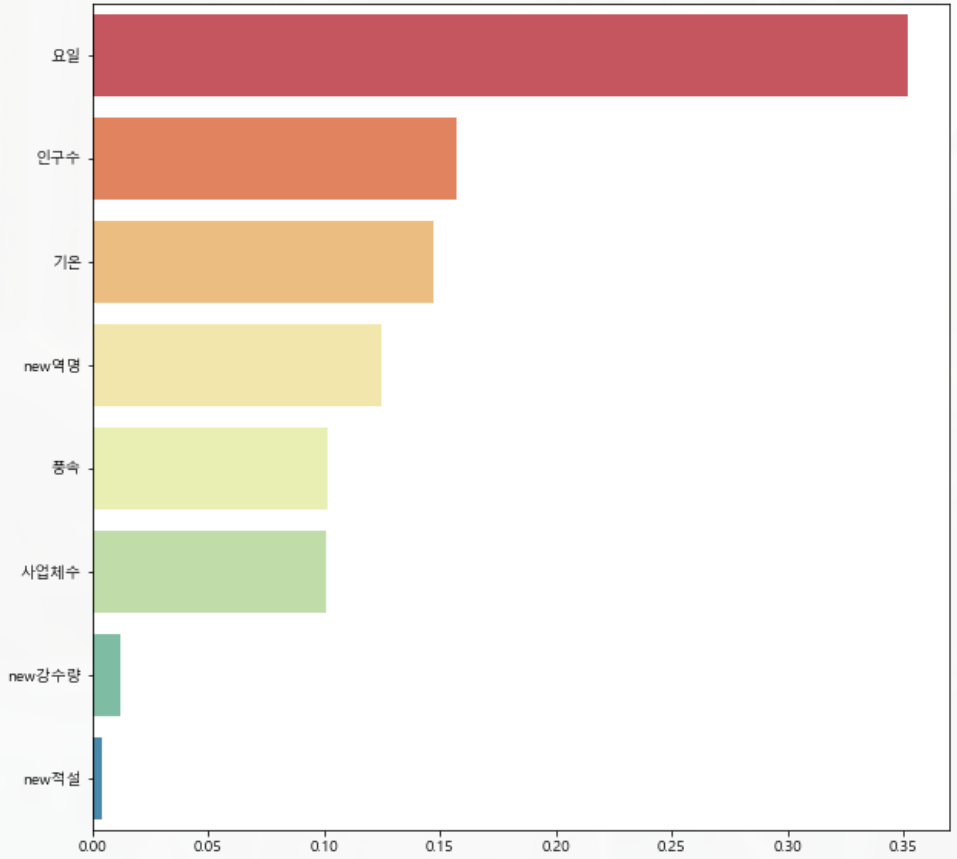




- Case1에서 변동성이 없는 데이터 중 중요도가 낮은 2개(관광지수, 상권수) 제외
- 가장 높은 성능은 RandomForest이고, Logistic Regression은 성능이 더 나빠졌다.
- Case 1(모든 변수)에 비교하여, Accuracy(+0.002)와 F1 Score(+0.007) 모두 상승했다.
- 즉, '관광지수와 상권수'가 모델의 성능을 오히려 저하할 수도 있다는 판단을 내릴 수 있다.

모델 성능 평가 (Score) – 계속

CASE 3 관광지수, 상권수 제외



[Best Model의 Feature Importance]

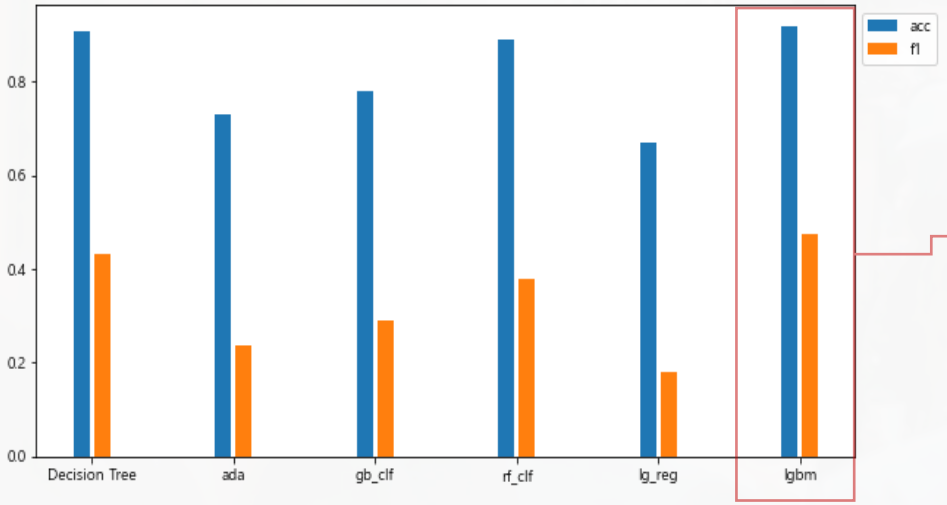
- 요일, 인구수, 기온 순으로 중요도가 높게 나타났다.
- 앞 모델들과 비교하면,
  1) (기온–인구수) → (인구수–기온) 순으로 변동
  2) (풍속–역명) → (역명–풍속) 순으로 변동
- 여전히 강수량과 적설의 중요도는 낮게 나타난다.

모델 성능 평가 (Score) – 계속

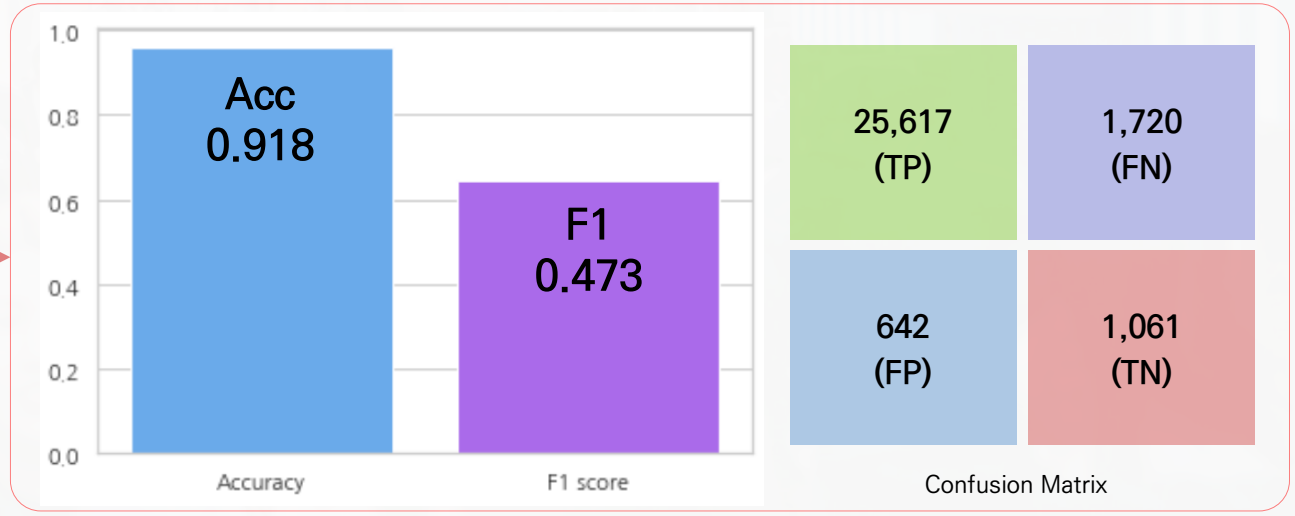
CASE 4 | 요일 제외

*상세 Score는 Appendix에서 확인 가능

[모델별 Accuracy / F1 Score]



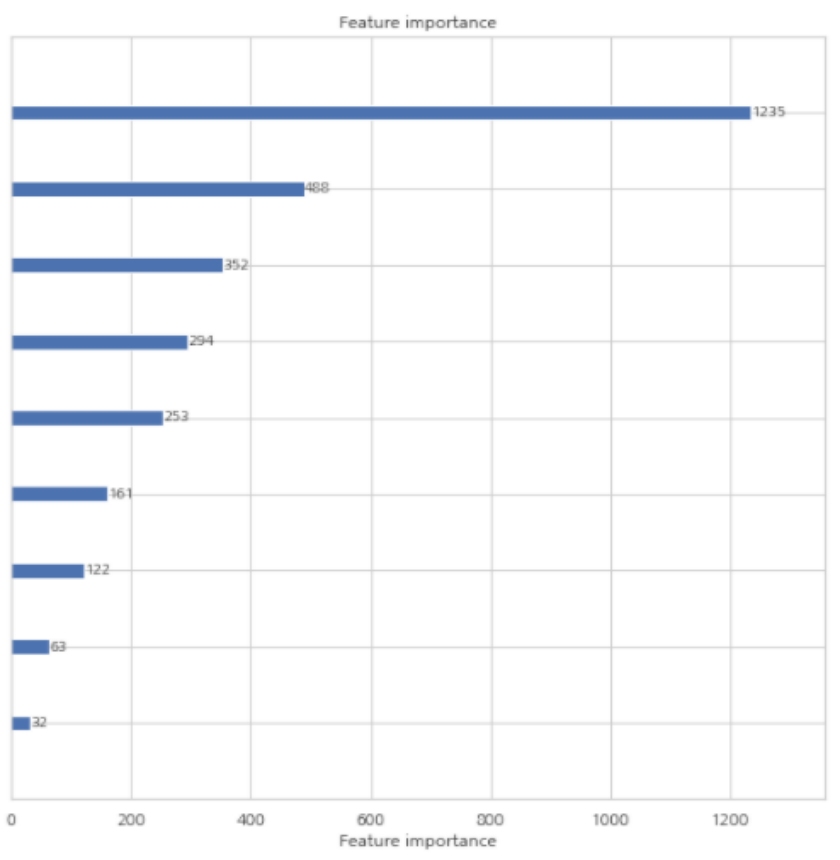
[Best Model (LGBM)]



- 주기성이 있는 특성(요일)을 제외할 시, Case1 대비 모델의 성능이 전체적으로 떨어진다.
- 가장 높은 성능은 LGBM으로 나타난다.
- 요일별 지하철 이용 목적이 크게 달라지기 때문에, '요일' 특성이 매우 중요한 영향을 미친다.

모델 성능 평가 (Score) – 계속

CASE 4 | 요일 제외



[Best Model의 Feature Importance]

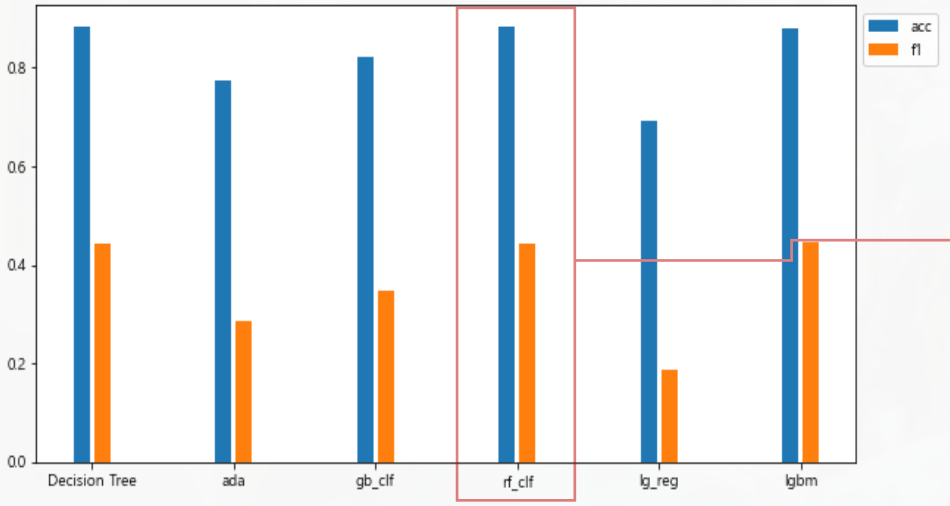
- 풍속, 기온, 역명 순으로 중요도가 높게 나타났다.
- 여전히 강수량과 적설의 중요도는 낮게 나타난다.

모델 성능 평가 (Score) – 계속

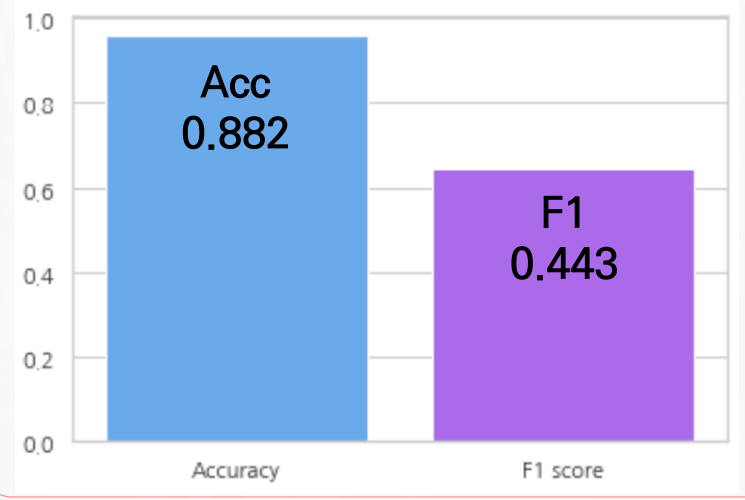
CASE 5 **날씨 전체 제외**

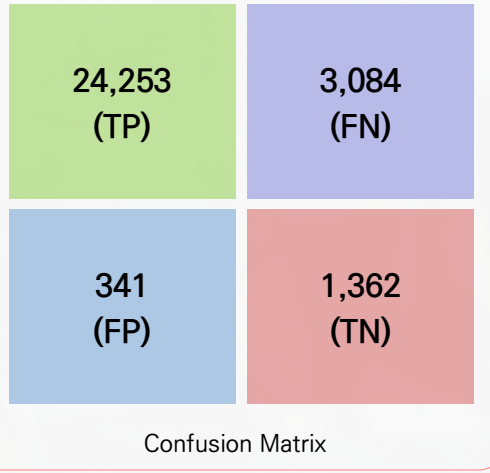
*상세 Score는 Appendix에서 확인 가능

[모델별 Accuracy / F1 Score]



[Best Model (RandomForest)]

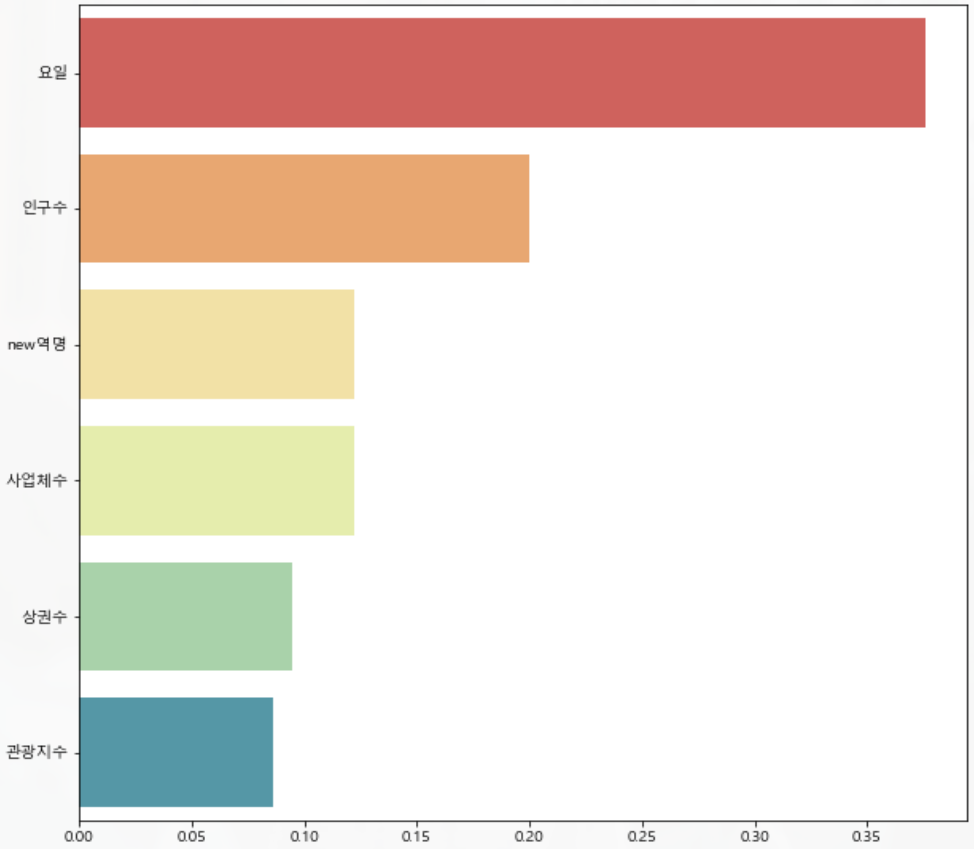




- 매일 변하는 특성(날씨)를 제외하면, Case1 대비 모델의 전체적인 성능은 나빠진다.
  - 날씨 특성 역시 모델의 성능에 영향을 주고 있다.

모델 성능 평가 (Score) – 계속

CASE 5 날씨 전체 제외



[Best Model의 Feature Importance]

- 요일, 인구수, 역명 순으로 중요도가 높게 나타났다.
- Case 1에서 날씨 Feature를 제외한 모든 중요도 순위가 일치한다.
- 사업체수 〉 상권수 〉 관광지수 순으로 중요도가 낮다.
- 대부분의 Feature가 비슷하게 반영되는 경향을 보인다.

최종 모델 (Final Model)

[혼잡 여부 예측 모델]

Classifier	RandomForestClassifier (Standard Scaler)
Feature	CASE 3 관광지수, 상권수 제외
Score	Acc : 0.959, F1 : 0.653
Reason	

- Acc, F1 - 가장 높은 점수
- 예상과는 다르게, 역 주변 관광지 수와 상권 수는 모델에 미치는 영향이 크지 않다고 판단
- 하차인원에 따른 혼잡 여부는, 방문 인원보다 근무자의 퇴근 영향을 받음
- 6~8시 만남을 위한 이동보다는, 대중교통 퇴근을 위한 이동을 고려해야 함

[Test] 미리 분리해둔 X_test 데이터로 확인

X_test

	요일	기온	풍속	new강수량	new적설	new역명	사업체수	인구수	상권수	관광지수
144204	4.0	-1.55	3.70	0	3	137	1737.0	102288.0	17.0	1.0
136900	3.0	1.30	3.55	0	3	42	5348.0	45731.0	28.0	1.0
71537	1.0	15.80	3.65	0	3	135	3317.0	177372.0	21.0	0.0
29894	4.0	14.15	1.50	0	3	140	633.0	26221.0	8.0	0.0
15727	6.0	-0.20	3.05	0	3	186	2895.0	120780.0	2.0	1.0
...	...	...	...	...	...	...	...	...	...	...

```
pred_test = rf_clf.predict(X_test)
wrong = X_test[y_test!=pred_test]
wrong.shape
```

```
(1541, 10)
```

▶ 36,300개 데이터 중 1,541개 틀림

5. 마무리

모델 → 서비스화

프로젝트 목표

내가 가려는 지하철역 인근이 혼잡할 지 예측한다.

MODEL

서비스

1) 내가 가려는 지하철역 인근이 혼잡할 확률 제시

ex) 강남(이)가 혼잡할 확률은 24%입니다.

2) 해당 인근 지역의 관광지를 함께 제시

ex) 강남 인근 관광지는 아래와 같습니다.

~~~~

서비스화를 위한 get_congestion 함수 정의

```
def get_congestion(day,temp,rain,snow,wind,station,hosun):
    train_label = df[df['역명']==station]['new역명'].unique()[0]
    company = df[(df['역명']==station)&(df['호선']==hosun)]['사업체수'].unique()[-1]
    p2m = visit[(visit['subway_station_nm']==station) & (visit['선명']==hosun)]['tourist_nm'].unique()
    people = people_now[(people_now['역명']==station) & (people_now['선명']==hosun)]['계'].unique()[0]
    final_model = Pipeline(estimators4)
    place = [day,temp,rain,snow,wind,train_label,company,people]
    result = final_model.predict_proba([place])

    if len(p2m):
        print(station + '인근 관광지는 아래와 같습니다.')
        print(p2m)
        print()
    else :
        print(station + '인근 관광지는 없습니다.')
    res = round(result[0][1]*100,2)
    return station+'(이)가 혼잡할 확률은 '+str(res)+'% 입니다.'
```

- 각 Feature 값을 넣으면 혼잡할 확률와 인근 관광지*가 출력됨

*인근 관광지 리스트는 서울시 지하철역별 관광지 데이터셋을 가공
~~~~

모델 → 서비스화 – 계속

✓ get_congestion(요일, 기온, 강수량, 적설, 풍속, 역명, 호선)

```
get_congestion(4, -4, 0, 0 , 7, '뚝섬','2호선')
```

```
뚝섬인근 관광지는 아래와 같습니다.
['브릭캠퍼스 서울' '서울숲']

'뚝섬가 혼잡할 확률은 41.0% 입니다.'
```

```
get_congestion(5, -11, 0, 0 , 11, '혜화','4호선')
```

```
혜화인근 관광지는 아래와 같습니다.
['국립서울과학관' '대학로 예술의 거리' '문예회관' '서울문묘' '이화마을' '짚, 풀 생활사 박물관' '창경궁']

'혜화가 혼잡할 확률은 12.0% 입니다.'
```

```
get_congestion(6, 3, 1.2, 0 , 2, '종각','1호선')
```

```
종각인근 관광지는 아래와 같습니다.
['관광안내전시관' '놀이동산 서울점' '런닝맨 인사동점' '보신각' '영풍문고' '조계사']

'종각가 혼잡할 확률은 13.0% 입니다.'
```

```
get_congestion(2, 30, 1.8, 0 , 4, '망원','6호선')
```

```
망원인근 관광지는 아래와 같습니다.
['망리단길' '망원 한강공원' '망원시장' '망원정터' '서교동최규하대통령가옥' '서울함 공원' '월드컵시장' '한강시민공원'
 '한강시민공원 망원지구 수영장']

'망원(이)가 혼잡할 확률은 14.0% 입니다.'
```

get_congestion() 예시

프로젝트 한계점

Cold Start Problem

위드코로나 데이터 부족
사업체수 등 변화 반영 미흡

*Cold Start Problem
모델 학습을 위한 충분한 데이터가
확보되어 있지 않아 모델 성능이 저하되는 문제

라벨링 문제

혼잡 여부 산출식의 주관성

데이터 불균형

데이터의 통일성 부족 – 전처리 어려움
결측치 처리에 따른 성능 저하

Appendix

모델 성능 평가 상세 내용

✓ Evaluation

CASE 1 모든 특성 선택

	acc	precision	recall	f1	roc_auc
Decision Tree	0.950999	0.950999	0.632413	0.602181	0.801629
ada	0.831990	0.831990	0.671756	0.319241	0.756864
gb_clf	0.879270	0.879270	0.686436	0.400068	0.788859
rf_clf	0.957610	0.957610	0.660012	0.646163	0.818081
lg_reg	0.669972	0.669972	0.622431	0.181135	0.647683
lgbm	0.947142	0.947142	0.749853	0.624603	0.854643

CASE 2 적설, 강수량 제외

	acc	precision	recall	f1	roc_auc
Decision Tree	0.950930	0.950930	0.627129	0.599832	0.799115
ada	0.836433	0.836433	0.696418	0.333053	0.770787
gb_clf	0.881612	0.881612	0.686436	0.404778	0.790103
rf_clf	0.954890	0.954890	0.665297	0.633669	0.819113
lg_reg	0.676825	0.676825	0.631826	0.186530	0.655727
lgbm	0.947796	0.947796	0.743981	0.625679	0.852237

CASE 3 관광지수, 상권수 제외

	acc	precision	recall	f1	roc_auc
Decision Tree	0.951963	0.951963	0.630652	0.606266	0.801316
ada	0.836019	0.836019	0.642983	0.315017	0.745514
gb_clf	0.868767	0.868767	0.668233	0.373912	0.774746
rf_clf	0.959470	0.959470	0.651791	0.653518	0.815214
lg_reg	0.581887	0.581887	0.614210	0.146972	0.597042
lgbm	0.948416	0.948416	0.751028	0.630671	0.855870

CASE 4 요일 제외

	acc	precision	recall	f1	roc_auc
Decision Tree	0.908884	0.908884	0.588373	0.430968	0.758612
ada	0.731543	0.731543	0.705226	0.235536	0.719204
gb_clf	0.779580	0.779580	0.774516	0.291846	0.777205
rf_clf	0.889463	0.889463	0.576629	0.379590	0.742790
lg_reg	0.670110	0.670110	0.622431	0.181197	0.647756
lgbm	0.918664	0.918664	0.623018	0.473238	0.780050

CASE 5 날씨 전체 제외

	acc	precision	recall	f1	roc_auc
Decision Tree	0.882782	0.882782	0.798591	0.444154	0.843309
ada	0.774174	0.774174	0.775690	0.287174	0.774885
gb_clf	0.820282	0.820282	0.813858	0.346890	0.817270
rf_clf	0.882369	0.882369	0.798591	0.443286	0.843089
lg_reg	0.689945	0.689945	0.608925	0.187218	0.651959
lgbm	0.878237	0.878237	0.832061	0.444898	0.856587


끝.