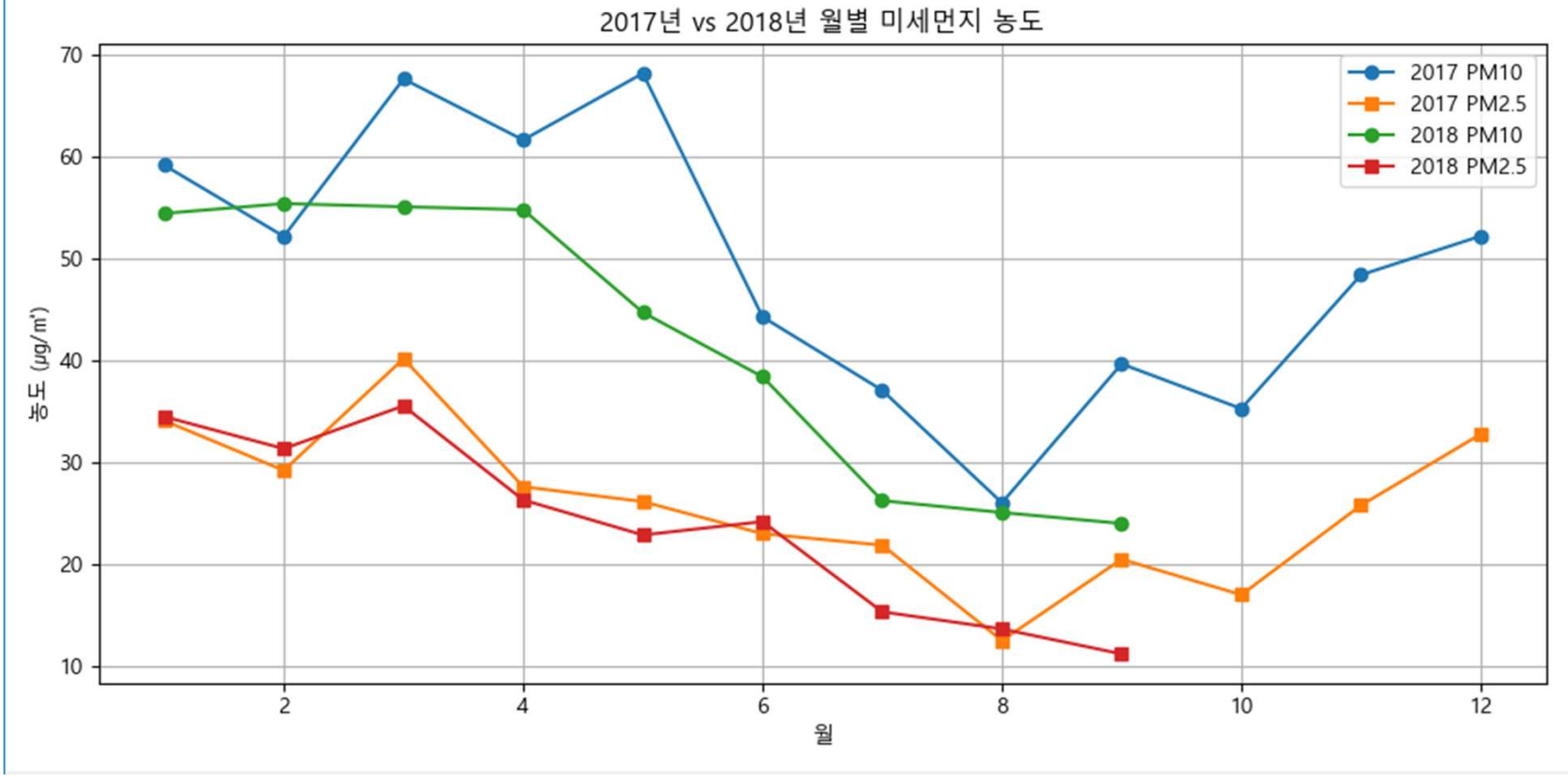
미세먼지 영향 예측을 활용한 마스크 수요 예측 및 프로모션 전략 고도화

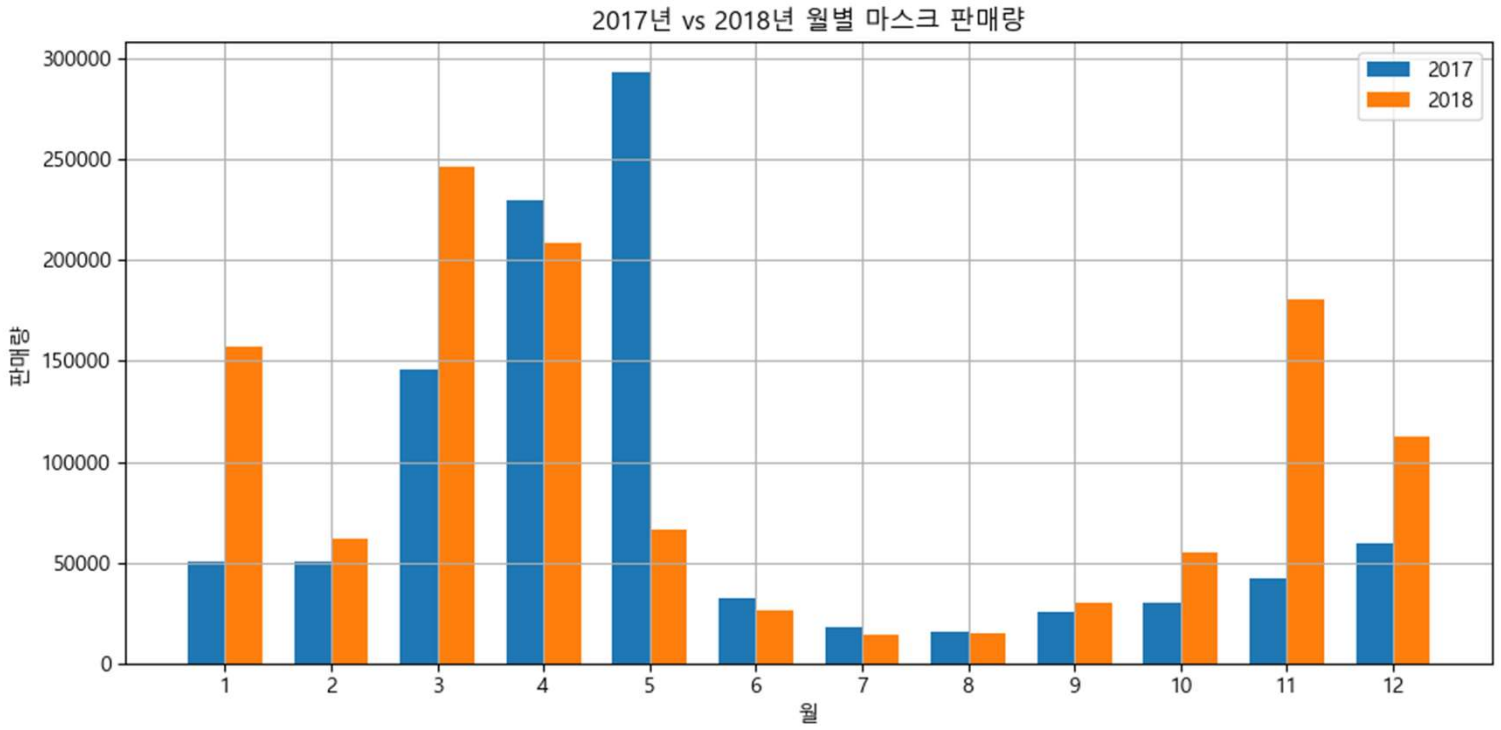
최준영

현상파악



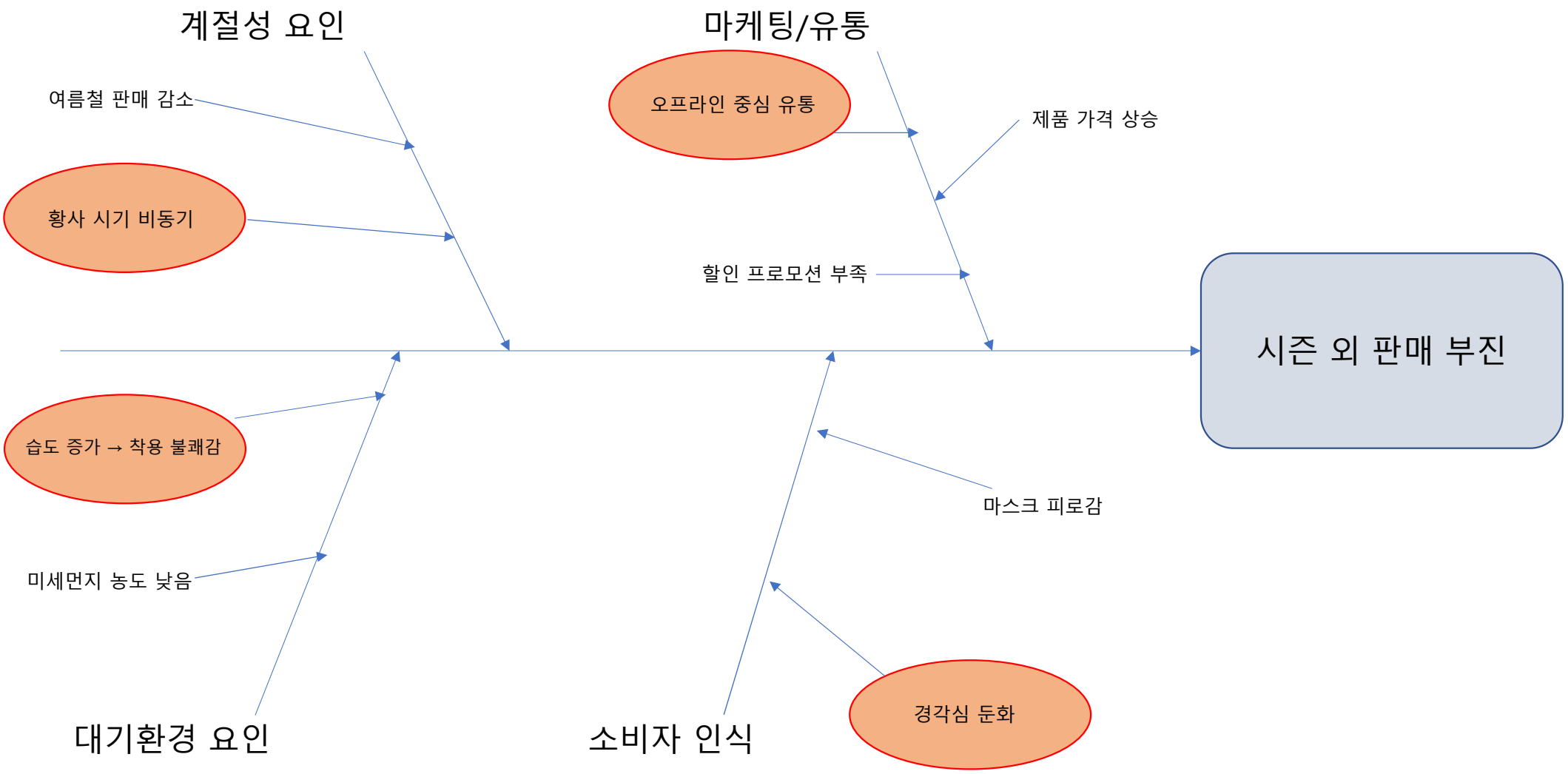
2018년에는 전반적으로 미세먼지(PM10, PM2.5) 농도가 감소했지만, 여전히 WHO 기준 (PM10 ≤ 15 μg/m³, Pm2.5 ≤ 5 μg/m³) 이상 수준이 유지되어 문제의 지속성이 나타남.

개선기회



2018년 마스크 수요가 유지되었으며, 초봄(1~3월)과 늦가을(11~12월) 등 계절성 강한 구간이 전체 판매량의 약 90%를 차지. 이러한 집중 수요 기간을 중심으로 기획된 마케팅·이벤트 전략이 효율적인 매출 증대 수단이 될 수 있음.

특성 요인도


계절성 요인
여름철 판매 감소
황사 시기 비동기
습도 증가 → 착용 불쾌감
미세먼지 농도 낮음
대기환경 요인
마케팅/유통
오프라인 중심 유통
제품 가격 상승
할인 프로모션 부족
소비자 인식
마스크 피로감
경각심 둔화
시즌 외 판매 부진

잠재 인자

잠재 원인	중요도	분석 가능성	합계	선정 여부
여름철 판매 감소	7	6	13	X
황사 시기 비동기	6	6	12	X
습도 증가 → 착용 불쾌감	8	5	13	X
미세먼지 농도 낮음	9	7	16	O
오프라인 중심 유통	9	7	16	O
제품 가격 상승	8	6	14	X
할인 프로모션 부족	7	7	14	X
경각심 둔화	9	8	17	O
마스크 피로감	7	6	13	X

분석 계획 수립

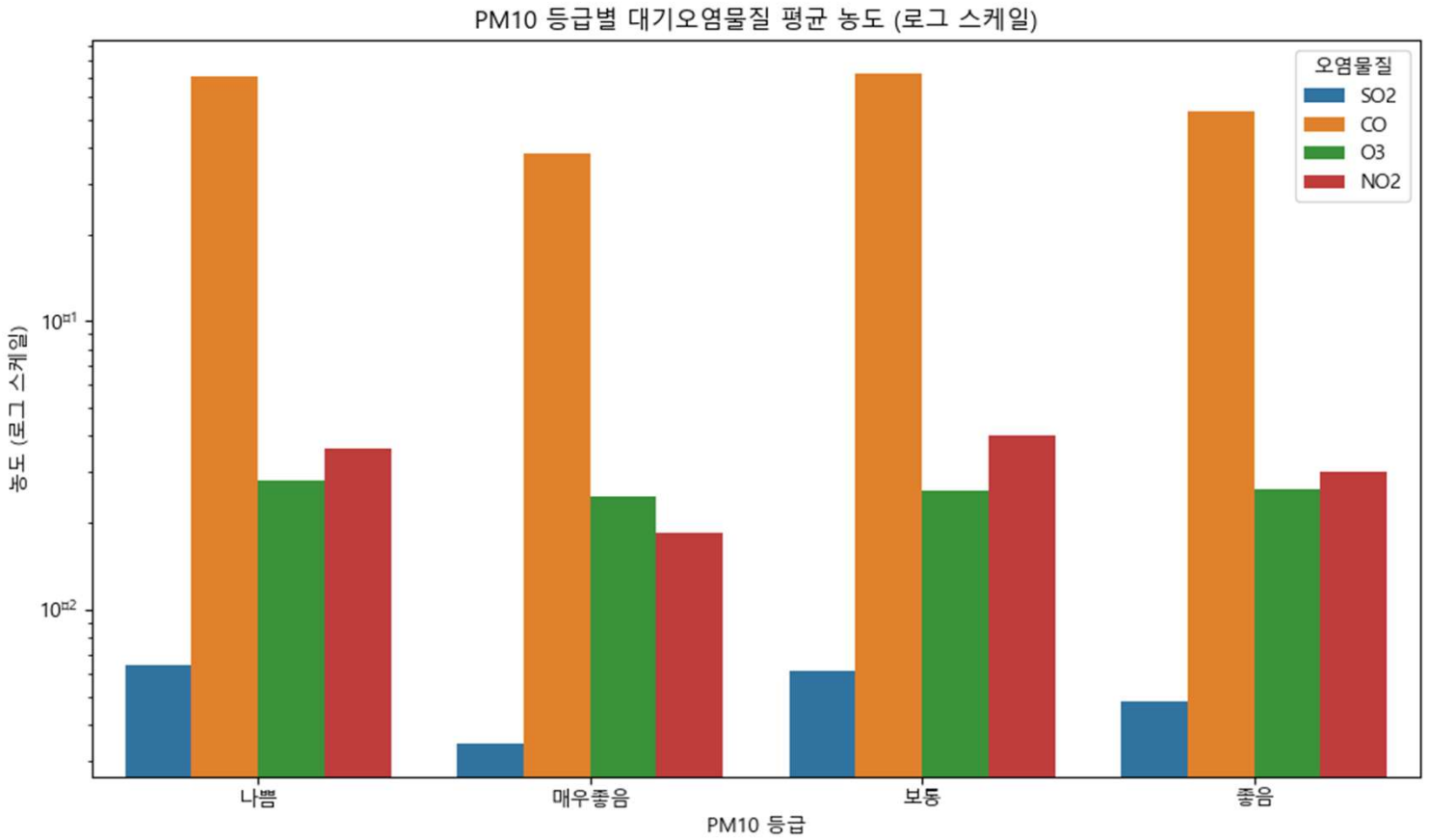
분석 주제	시각화 방식	목적 / 기대효과
월별 판매량 추이	막대차트	계절성 파악
미세먼지 vs 판매량	산점도 + 회귀	민감도 확인 및 수요 변화 예측 가능성
잠재 요인 상관분석	히트맵	우선 개입 요인 도출
시즌 외 판매 부진 패턴	조건 필터링	취약 기간 및 요인 정밀 분석

데이터 분석 – 습도 범주별 평균 판매건수 비교

습도 및 대기오염도와 마스크 수요 간에는 뚜렷한 상관성이 관찰되었다.

- 습도가 낮은 날일수록 마스크 판매량이 증가, 습도가 높은 날일수록 마스크 판매량 감소

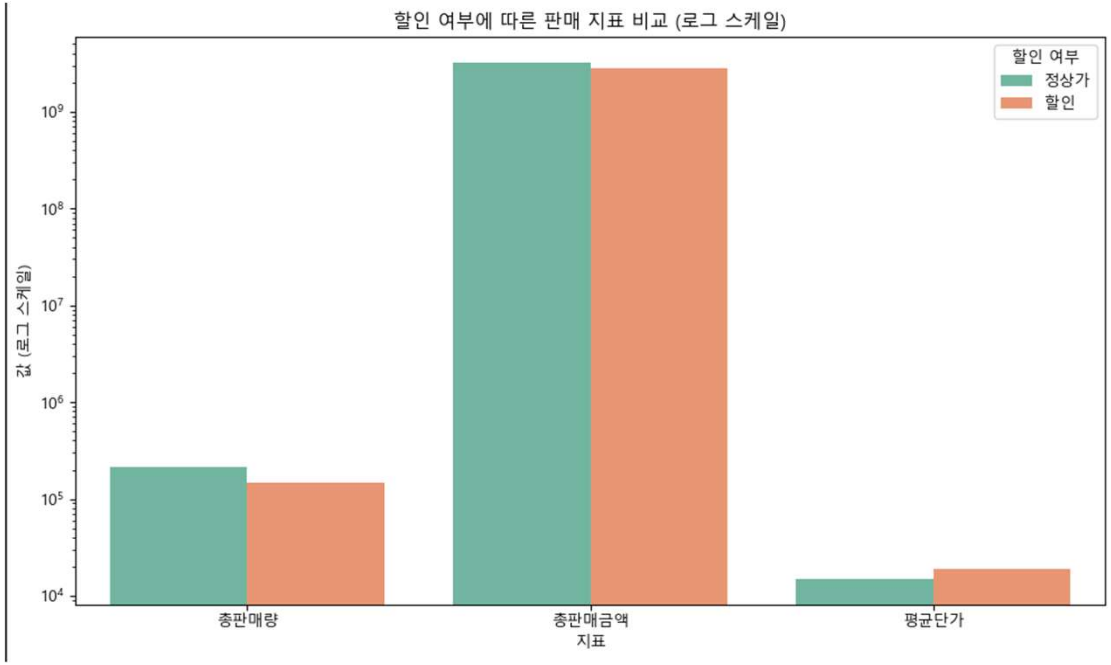
데이터 분석 – PM10 등급별 대기오염물질 비교

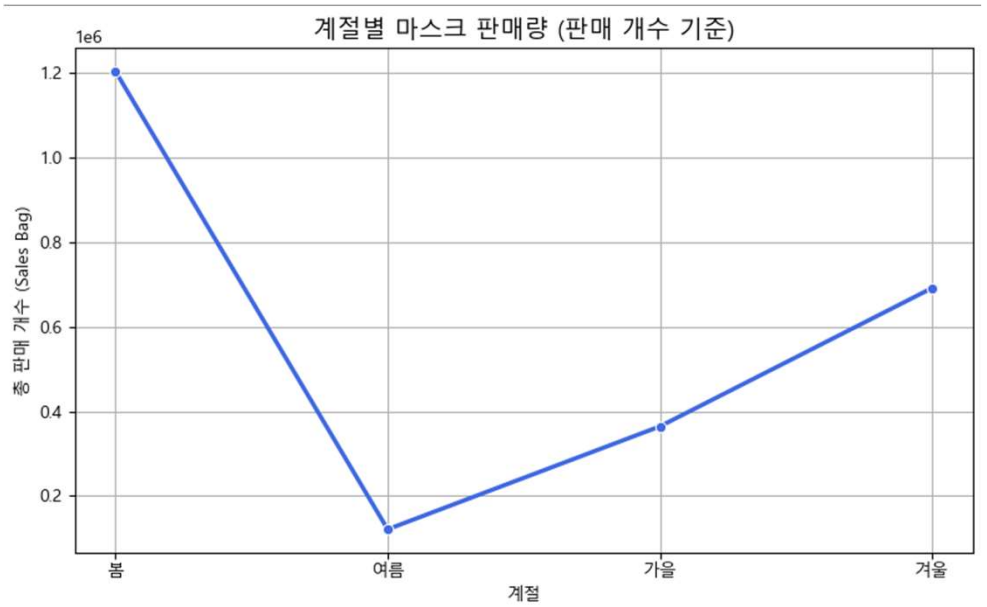


PM10 등급이 나쁠수록 CO, NO_2 등의 대기오염물질 농도가 함께 상승하는 경향이 확인되었으며, 이는 복합적인 오염의 동시 발생 가능성을 시사

- PM10과 NO_2, CO의 상관성이 높은 것으로 확인됨

데이터 분석 – 마스크 판매 지표 요인별 비교

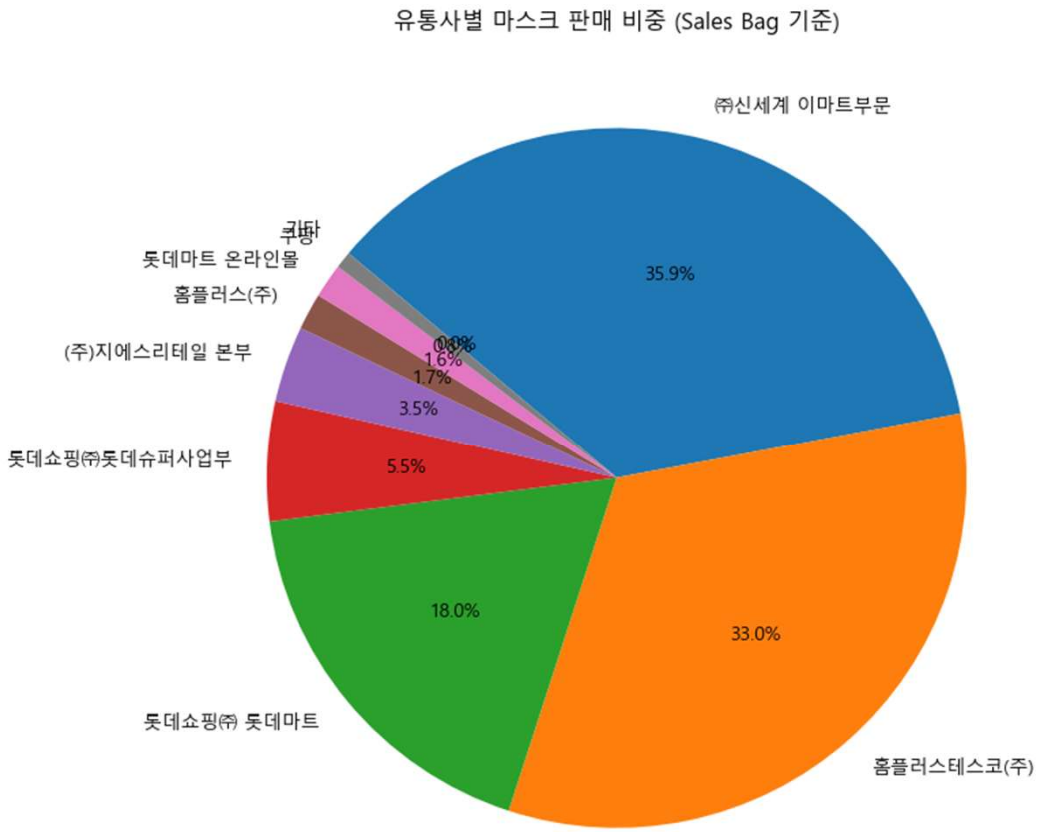




마스크 판매량은 계절이라는 변수에 영향을 크게 받는 것으로 확인되며, 할인이 판매 지표에 긍정적 영향을 줄 수 있는 것으로 확인됨

데이터 분석 – 마스크 판매 지표 요인별 비교

이마트와 홈플러스가 전체 마스크 판매의 약 70% 이상을 차지하여 신규 유통 채널 발굴 또는 중소 유통사의 경쟁력 강화 전략 수립 시 참고 가능



데이터 분석 – 미세먼지, 날씨와 마스크 판매량의 관계

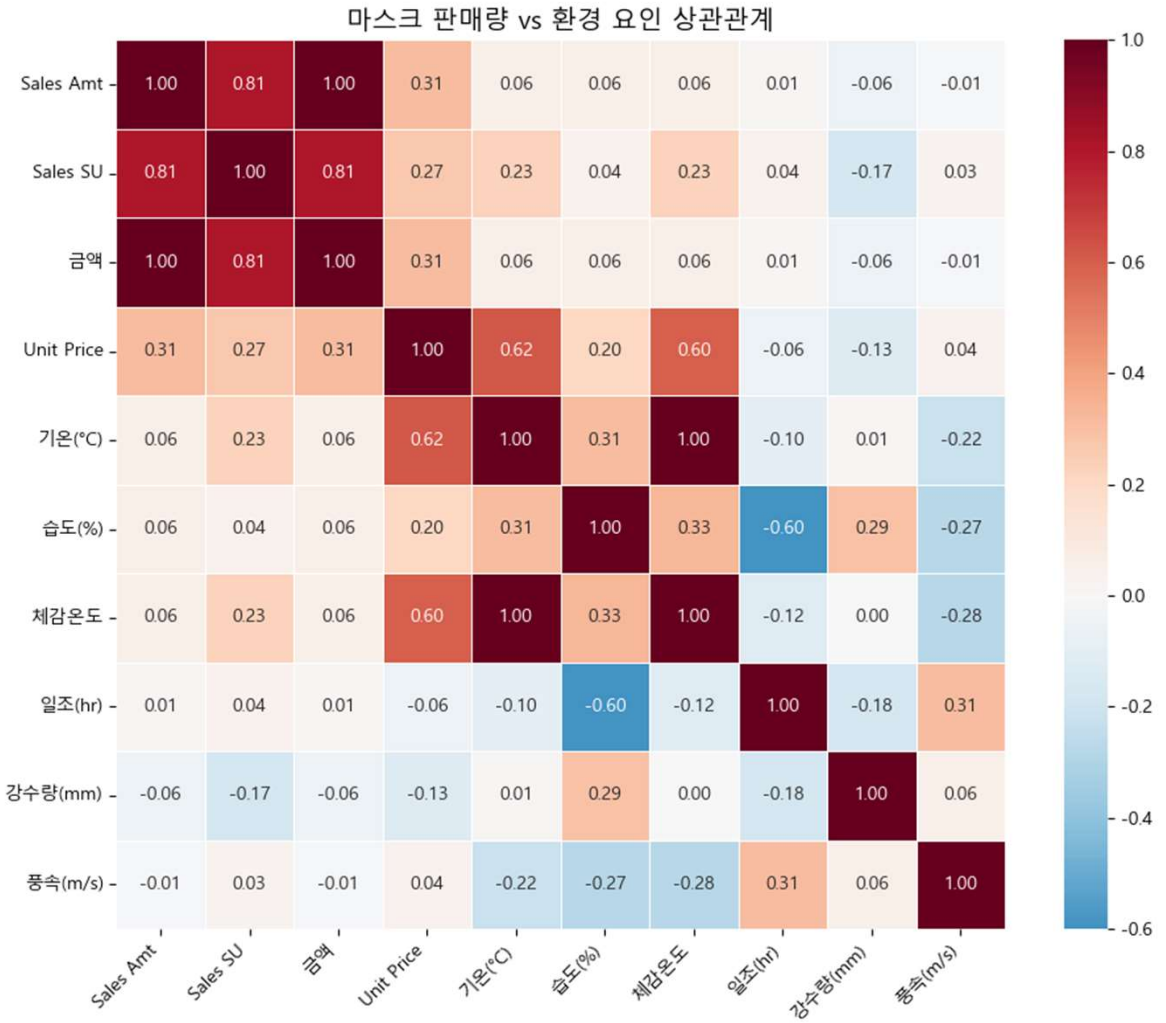
PM 10, PM 2.5와 마스크 판매량(금액, 수량) 간에 높은 양의 상관관계 존재(0.7 ~ 0.8 수준)

기온(℃), 체감온도와는 낮은 음의 상관관계

일조시간, 풍속 등은 영향 미미 / 습도는 거의 무관

CO, NO2 등 기체 오염물질과 판매량은 약한 관계

핵심 결론: 미세먼지는 마스크 수요의 중요한 예측 변수로 활용 가능



머신러닝 모델링 – 데이터 준비

```
dust_daily = dust_df.groupby(dust_df['날짜'].dt.date).mean(numeric_only=True).reset_index()
dust_daily['날짜'] = pd.to_datetime(dust_daily['날짜'])

weather_daily = weather_df.groupby(weather_df['날짜'].dt.date).mean(numeric_only=True).reset_index()
weather_daily['날짜'] = pd.to_datetime(weather_daily['날짜'])

sales_daily = sales_df.groupby(sales_df['날짜'].dt.date).sum(numeric_only=True).reset_index()
sales_daily['날짜'] = pd.to_datetime(sales_daily['날짜'])
```

데이터를 시간 기반으로 예측이 가능하게 변환

```
# 3. 병합 및 파생 변수
merged = pd.merge(sales_daily, dust_daily, on='날짜', how='inner')
merged = pd.merge(merged, weather_daily, on='날짜', how='inner')
```

다양한 소스를 통합하여 예측에 필요한 환경 데이터를 구성하여 모델의 성능을 끌어올리고자 하였음

머신러닝 모델링 – Feature Engineering

성능 향상을 위해 의미있는 변수들을 추가 생성

이상치 제거 및 로그 변환으로 예측 가능성 증대

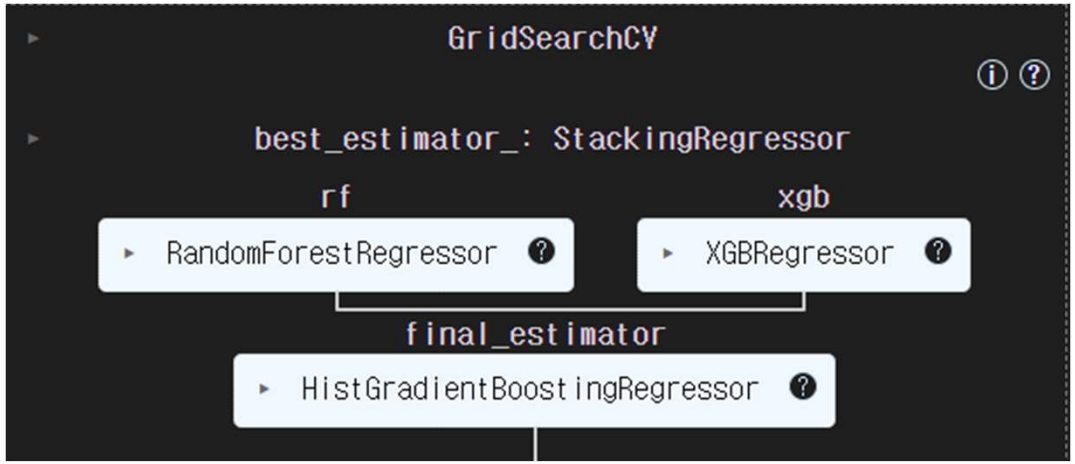
실제 판매량 변동성을 모델에 적용하게 됨

```
merged = pd.merge(sales_daily, dust_daily, on='날짜', how='inner')
merged = pd.merge(merged, weather_daily, on='날짜', how='inner')
merged['체감온도'] = merged['기온(°C)'] - 0.7 * merged['풍속(m/s)'] + 0.03 * merged['습도(%)']
merged['기압차'] = merged['해면기압(hPa)'] - merged['현지기압(hPa)']
merged['월'] = merged['날짜'].dt.month
merged['요일'] = merged['날짜'].dt.dayofweek
merged['풍압'] = merged['풍속(m/s)'] * merged['풍향(16방위)']
merged['판매량_어제'] = merged['Sales Amt'].shift(1)
merged['판매량_7일평균'] = merged['Sales Amt'].rolling(window=7).mean()
merged = merged.dropna(subset=['판매량_어제', '판매량_7일평균']) # dropna를 전체 데이터프레임에 적용하지

# 이상치 제거 + 로그 변환
q1 = merged['Sales Amt'].quantile(0.25)
q3 = merged['Sales Amt'].quantile(0.75)
iqr = q3 - q1
merged = merged[(merged['Sales Amt'] >= q1 - 1.5 * iqr) & (merged['Sales Amt'] <= q3 + 1.5 * iqr)]
merged['Sales Amt'] = np.log1p(merged['Sales Amt'])

# 카테고리형 변수 원-핫 인코딩
merged = pd.get_dummies(merged, columns=['요일', '월'], drop_first=True)
```

머신러닝 모델링 – 앙상블 모델 구성

랜덤 포레스트(RandomForest) + XGBoost를 base 모델로 한 스태킹(StackongRegressor) 앙상블 모델 구성하여 각 모델들의 강점을 결합하여 예측 성능 극대화

최종 메타 모델로는 HistGradientBoostingRegressor를 사용하여 다양한 모델의 결과를 학습하게 해 더욱 정교한 예측을 도출하는 앙상블 전략 사용



머신러닝 모델링 – 하이퍼파라미터 튜닝

주요 하이퍼파라미터 조합을 GridSearchCV로 탐색하며 모델 성능을 최적화하기 위해 다양한 파라미터 조합을 테스트

5-fold 교차검증(CV)를 통해 과적합 방지 및 일반화 성능을 확보하며 안정적인 모델을 선택

```python
param_grid = {
    'rf__n_estimators': [200, 300, 500],
    'rf__max_depth': [None, 10, 20],
    'xgb__n_estimators': [200, 300],
    'xgb__learning_rate': [0.03, 0.05, 0.07],
    'xgb__max_depth': [3, 5, 7]
}

grid = GridSearchCV(stack, param_grid, cv=3, scoring='r2', n_jobs=-1)
grid.fit(X_train, y_train)

cv_score = cross_val_score(grid.best_estimator_, X_train, y_train, cv=5, scoring='r2')
```

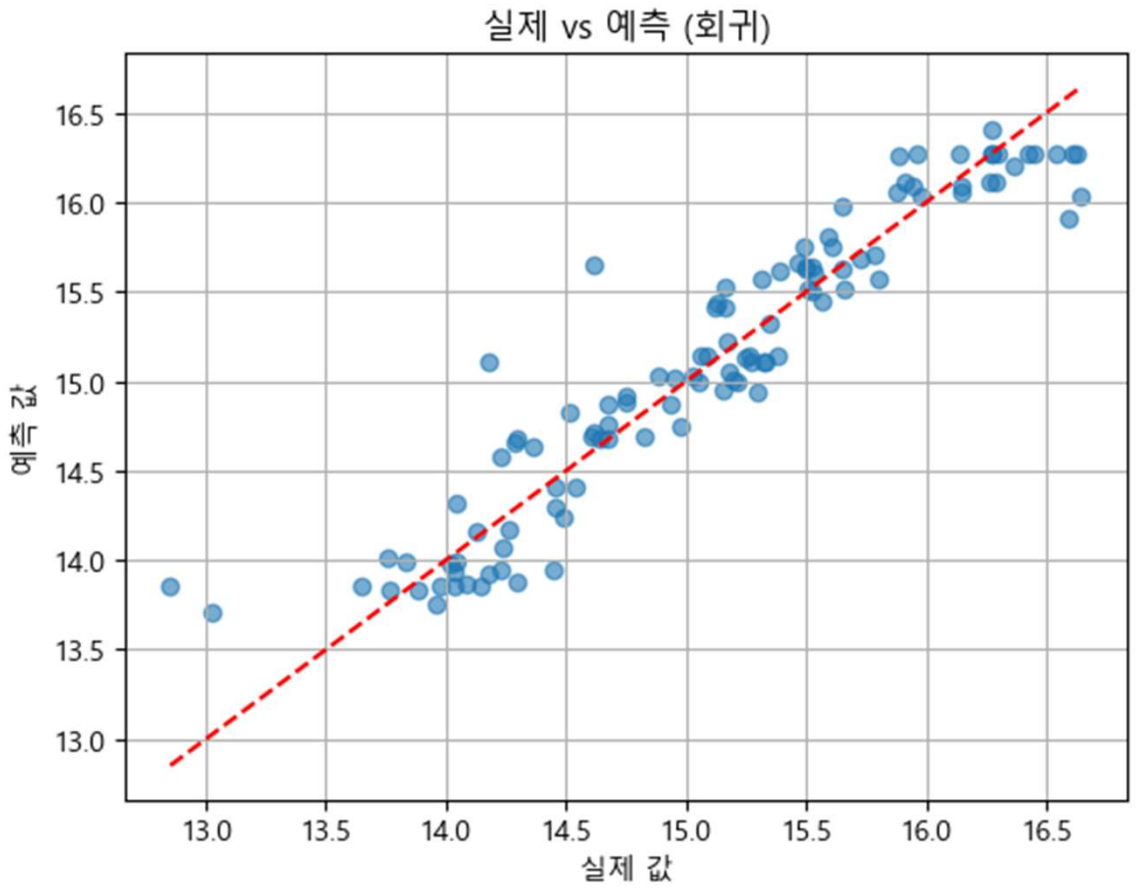
머신러닝 모델링 – 평가 및 결과 해석

R^2 Score, RMSE, MAE의 수치가 각각 0.8910, 0.28, 0.20 으로 준수한 성능의 모델이 완성됨

추가적으로 모델에 기여도가 낮은 Feature들 중 상위 10개를 출력하여 이후 해당 변수들을 제거하거나 개선할 여지를 판단할 수 있게 함

```
✅ R² Score : 0.8910
✅ RMSE     : 0.28
✅ MAE      : 0.20
✅ 최적 하이퍼파라미터: {'rf__max_depth': 10, 'rf__n_estimators': 300, 'xgb__learning_rate': 0.03, 'xgb__max_depth': 3, 'xgb__n_estimators': 200}
✅ 평균 CV R² Score: 0.7914
Base estimators: {'rf': RandomForestRegressor(max_depth=10, n_estimators=300, random_state=42), 'xgb': XGBRegressor(base_score=None, booster=None, callbacl
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, device=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None, feature_types=None,
             feature_weights=None, gamma=None, grow_policy=None,
             importance_type=None, interaction_constraints=None,
             learning_rate=0.03, max_bin=None, max_cat_threshold=None,
             max_cat_to_onehot=None, max_delta_step=None, max_depth=3,
             max_leaves=None, min_child_weight=None, missing=nan,
             monotone_constraints=None, multi_strategy=None, n_estimators=200,
             n_jobs=None, num_parallel_tree=None, ...)}

📉 하위 기여도 Feature:
   Feature  Importance
33     월_8    0.000052
36    월_11    0.000060
32     월_7    0.000138
30     월_5    0.000151
35    월_10    0.000158
37    월_12    0.000169
31     월_6    0.000204
23    요일_3    0.000239
29     월_4    0.000262
27     월_2    0.000315
```

머신러닝 모델링 – 개선



위 그래프는 모델의 분류 결과에 대한 시각화로, 몇 개의 이상치를 제외하면 모델의 예측 결과가 실제 값에 근접하게 다가가고 있다는 것을 확인 가능

머신러닝 모델링 – 개선

✅ 모델이 잘 예측한 샘플 (오차가 거의 없음):

	PM10	PM25	풍속(m/s)	풍향(16방위)	습도(%)	증기압(hPa)	\
300	53.725653	30.671642	1.983333	152.500000	52.083333	9.758333	
264	41.198923	23.552595	2.141667	147.083333	60.041667	14.170833	

	이슬점온도(°C)	현지기압(hPa)	해면기압(hPa)	최저운고(100m)	...	월_6	월_7	월_8	\
300	6.570833	1010.000000	1020.1375	17.788291	...	False	False	False	
264	12.066667	1002.854167	1012.8000	14.400000	...	False	False	False	

	월_9	월_10	월_11	월_12	실제값	예측값	오차
300	False	True	False	False	15.024281	15.023182	0.001099
264	True	False	False	False	14.672596	14.676623	0.004027

❌ 모델이 잘못 예측한 샘플 (오차 큼):

	PM10	PM25	풍속(m/s)	풍향(16방위)	습도(%)	증기압(hPa)	\
21	80.977552	27.682048	3.587500	282.916667	52.833333	1.850000	
567	33.106483	20.486412	1.395833	192.083333	50.916667	22.641667	
357	78.640156	60.934638	1.825000	190.000000	82.791667	6.575000	
80	58.107132	30.780976	2.525000	280.000000	33.458333	3.258333	
203	53.726155	36.183238	2.129167	212.916667	89.041667	31.741667	

	이슬점온도(°C)	현지기압(hPa)	해면기압(hPa)	최저운고(100m)	...	월_6	월_7	\
21	-15.683333	1016.145833	1027.379167	8.333333	...	False	False	
567	19.458333	998.591667	1008.104167	61.500000	...	False	True	
357	0.708333	1003.529167	1014.170833	5.550000	...	False	False	
80	-8.375000	1010.129167	1020.670833	58.666667	...	False	False	
203	25.008333	995.091667	1004.716667	6.208333	...	False	True	

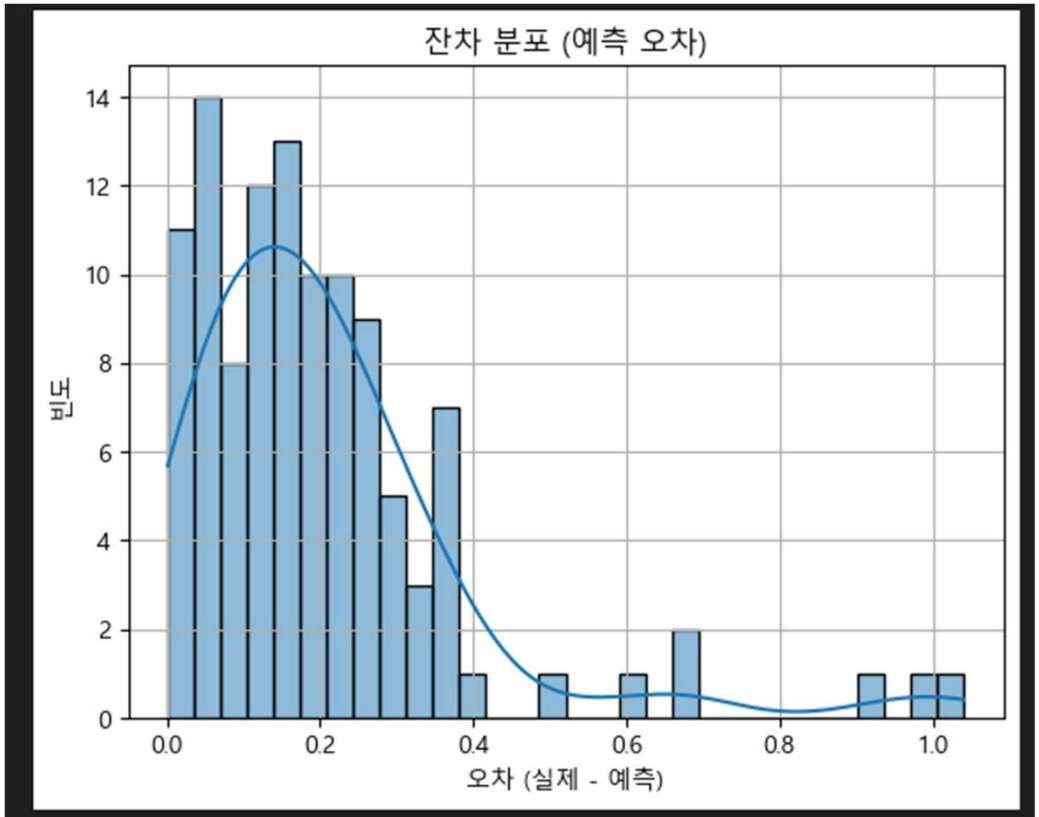
	월_8	월_9	월_10	월_11	월_12	실제값	예측값	오차
21	False	False	False	False	False	14.611453	15.651266	1.039813
567	False	False	False	False	False	12.852285	13.849807	0.997521
357	False	False	False	False	True	14.176990	15.107940	0.930950
80	False	False	False	False	False	16.591036	15.903940	0.687096
203	False	False	False	False	False	13.029320	13.705257	0.675936

모델이 잘 예측한 데이터의 오차는 거의 없으나 모델이 잘못 예측한 데이터의 오차는 매우 큰 것으로 확인됨

머신러닝 모델링 – 개선

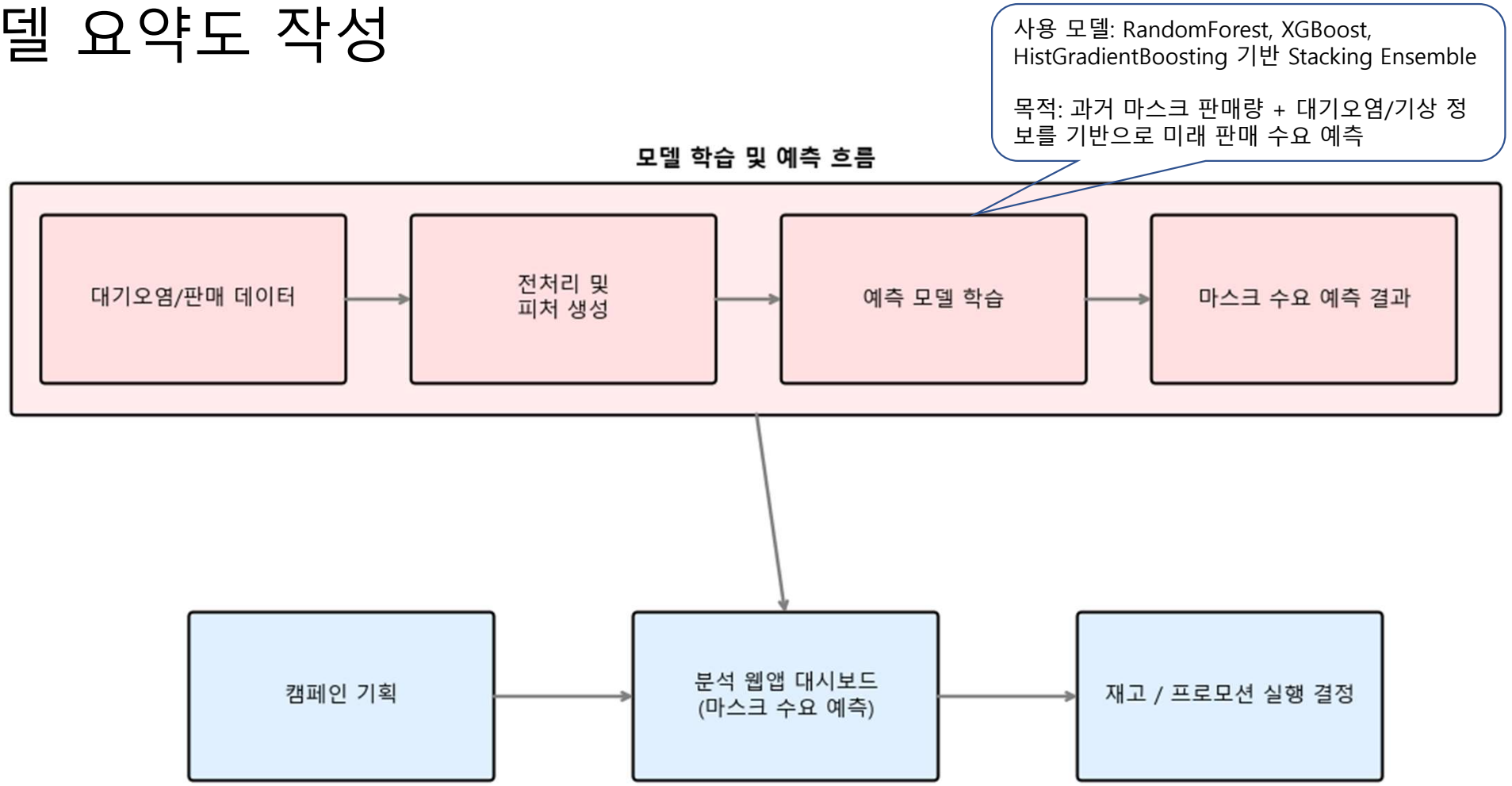
오차의 분포를 히스토그램으로 시각화 하였고, 이와 상관관계가 높은 변수가 시계열 데이터인 요일과 월이었음을 확인했다.

이 부분은 시계열 특화 모델 적용(예: LSTM) 등의 방법으로 오차를 줄일 수 있을 것으로 예상된다.




```
📊 오차와 상관관계가 높은 변수:
요일_6    0.509644
월_7      0.193348
Name: 오차, dtype: float64
```

모델 요약도 작성


사용 모델: RandomForest, XGBoost, HistGradientBoosting 기반 Stacking Ensemble
목적: 과거 마스크 판매량 + 대기오염/기상 정보를 기반으로 미래 판매 수요 예측
모델 학습 및 예측 흐름
대기오염/판매 데이터
전처리 및 피처 생성
예측 모델 학습
마스크 수요 예측 결과
캠페인 기획
분석 웹앱 대시보드 (마스크 수요 예측)
재고 / 프로모션 실행 결정

해결 방안 및 기대 효과

해결 방안: 비지도학습 확장(추후 적용 가능)

- 소비자 패턴 군집화 -> 마케팅 타겟팅
- 이상치 탐지 -> 예외 상황 모니터링

기대효과

- 수요 예측 정확도 향상(재고 과잉 및 부족 방지 및 마스크 공급 안정화)
- 재고 운영 효율화(날씨/오염도 기반 사전 재고 이동 가능)
- 마케팅 최적화(고수요 예측 기간에 프로모션 집중 가능)
- 비용 절감(유통/물류 최소화 및 폐기 비용 절감)

실제 적용 방안

1.모델 학습 및 예측 파이프라인

- [대기오염 + 판매 데이터] -> [전처리 및 파생 변수 생성] -> [예측모델 학습] -> [미래 마스크 수요 예측 결과 도출]

2.분석 결과 활용

- 예측 결과는 분석 웹앱 대시보드에 자동 시각화 됨
- 이를 통해
  - 캠페인 기획팀은 고수요 기간/지역 중심의 프로모션 계획 수립
  - 물류/운영팀은 지역별 재고 사전배치 및 긴급 공급 계획 수립

streamlit 예시