



ML Session Final Project

일상 속 외부 요인을 고려한 매출 예측 모델 설계

C조

고지원 정유환 최정연 표수영

CONTENTS



1 Intro

1-01 주제

1-02 선정 배경 및 목적

2 Data

2-01 Dataset

2-02 데이터 전처리

2-03 데이터 시각화

3 Modeling

3-01 Linear Regression

3-02 Random Forest

3-03 XGBoost

3-04 Ensemble

4 Analysis

4-01 비교 분석

4-02 한계 및 개선 방안

일상 속 외부 요인을 고려한 매출 예측 모델 설계

#비오는날엔_파전 #불금 날씨, 휴일 등 외부 요인에 영향을 받는 소비 경향

기온, 휴일, 실업률, CPI, 유가 등의 외부 요인이 매출에 미치는 영향을 분석하고 예측 모델 구축




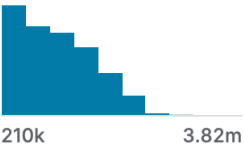

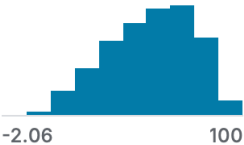

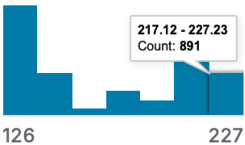

Walmart 주간 매출 데이터

<https://www.kaggle.com/datasets/mikhail1681/walmart-sales>

공휴일, 경제 지표(실업률, CPI 등), 날씨 요소(온도), 유가 정보가 함께 포함되어 있어 다양한 외부 요인이 매출에 어떤 영향을 미치는지를 분석할 수 있도록 구성

Walmart_Sales.csv (363.73 kB)

Detail Compact Column

# Store Store number	# Date Sales week start date	# Weekly_Sales Sales	# Holiday_Flag Mark on the presence or absence of a holiday	# Temperature Air temperature in the region	# Fuel_Price Fuel cost in the region	# CPI Consumer price index	# Unemployment Unemployment rate
 1 45	143 unique values	 210k 3.82m	 0 1	 -2.06 100	 2.47 4.47	 126 227 217.12 - 227.23 Count: 891	 3.88 14.3
1	05-02-2010	1643690.9	0	42.31	2.572	211.0963582	8.106
1	12-02-2010	1641957.44	1	38.51	2.548	211.2421698	8.106
1	19-02-2010	1611968.17	0	39.93	2.514	211.2891429	8.106
1	26-02-2010	1409727.59	0	46.63	2.561	211.3196429	8.106
1	05-03-2010	1554806.68	0	46.5	2.625	211.3501429	8.106

▶ 결측치

```
>>> <class 'pandas.core.frame.DataFrame'>
RangeIndex: 6435 entries, 0 to 6434
Data columns (total 8 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Store            6435 non-null   int64
1   Date             6435 non-null   object
2   Weekly_Sales     6435 non-null   float64
3   Holiday_Flag     6435 non-null   int64
4   Temperature      6435 non-null   float64
5   Fuel_Price       6435 non-null   float64
6   CPI              6435 non-null   float64
7   Unemployment     6435 non-null   float64
dtypes: float64(5), int64(2), object(1)
```

결측치 없음

▶ 정규화, 원-핫 인코딩 (One-Hot Encoding)

```
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline

preprocessor = ColumnTransformer([
    ('num', MinMaxScaler(), ['Temperature', 'Fuel_Price', 'CPI', 'Unemployment']),
    ('cat', OneHotEncoder(drop='first', sparse_output=False),
     ['DayOfWeek', 'Holiday_Flag']),
])
```

▶ 이상치

```
q1 = df['Weekly_Sales'].quantile(0.25)
q3 = df['Weekly_Sales'].quantile(0.75)
iqr = q3 - q1
lower, upper = q1 - 1.5 * iqr, q3 + 1.5 * iqr

outliers_lower = (df['Weekly_Sales'] < lower).sum()
outliers_upper = (df['Weekly_Sales'] > upper).sum()

print(f'Lower outliers count: {outliers_lower}')
print(f'Upper outliers count: {outliers_upper}')
```

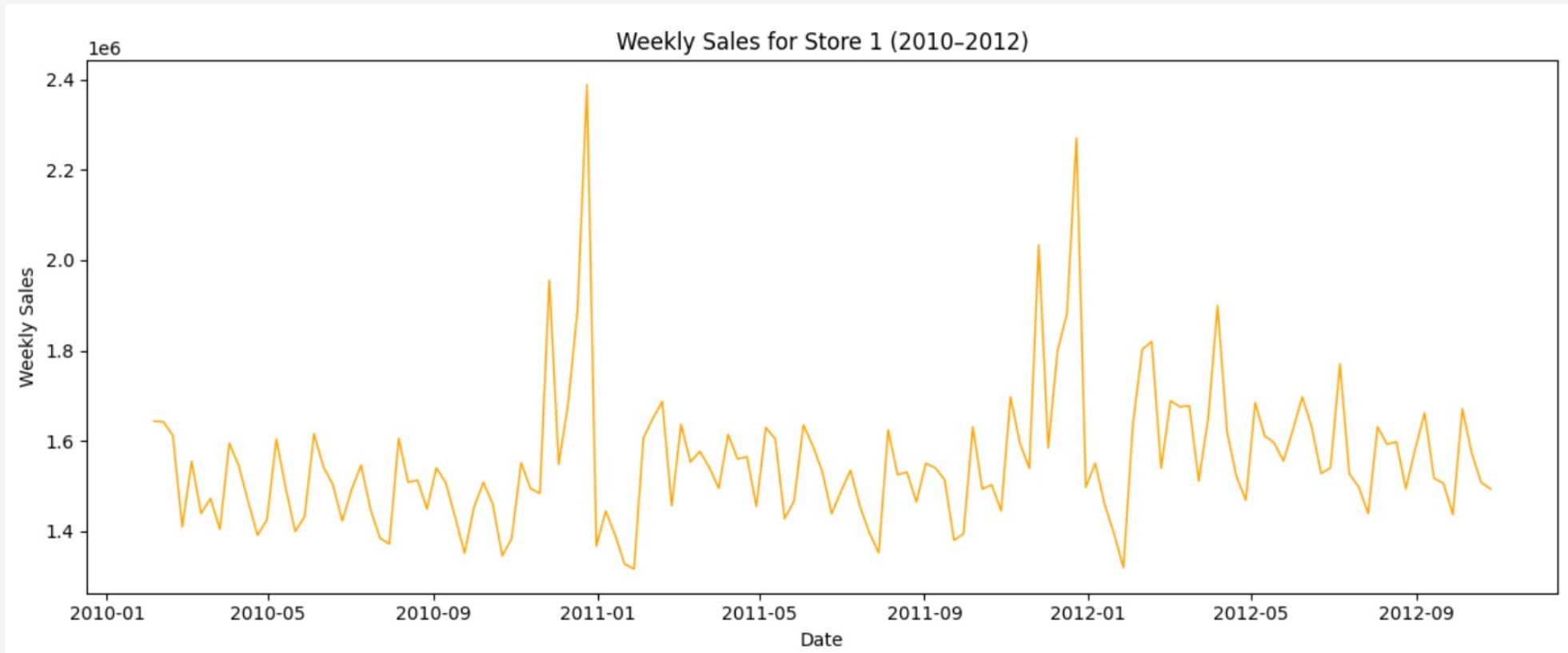
```
>>> Lower outliers count: 0
Upper outliers count: 0
```

이상치 없음

▶ 날짜 파싱 (parsing)

```
def parse_and_engineer(df: pd.DataFrame) -> pd.DataFrame:
    df = df.copy()
    df['Date'] = pd.to_datetime(df['Date'], dayfirst=True)
    df['Year'] = df['Date'].dt.year
    df['Month'] = df['Date'].dt.month
    df['Week'] = df['Date'].dt.isocalendar().week
    df['DayOfWeek'] = df['Date'].dt.day_name()
    return df
```

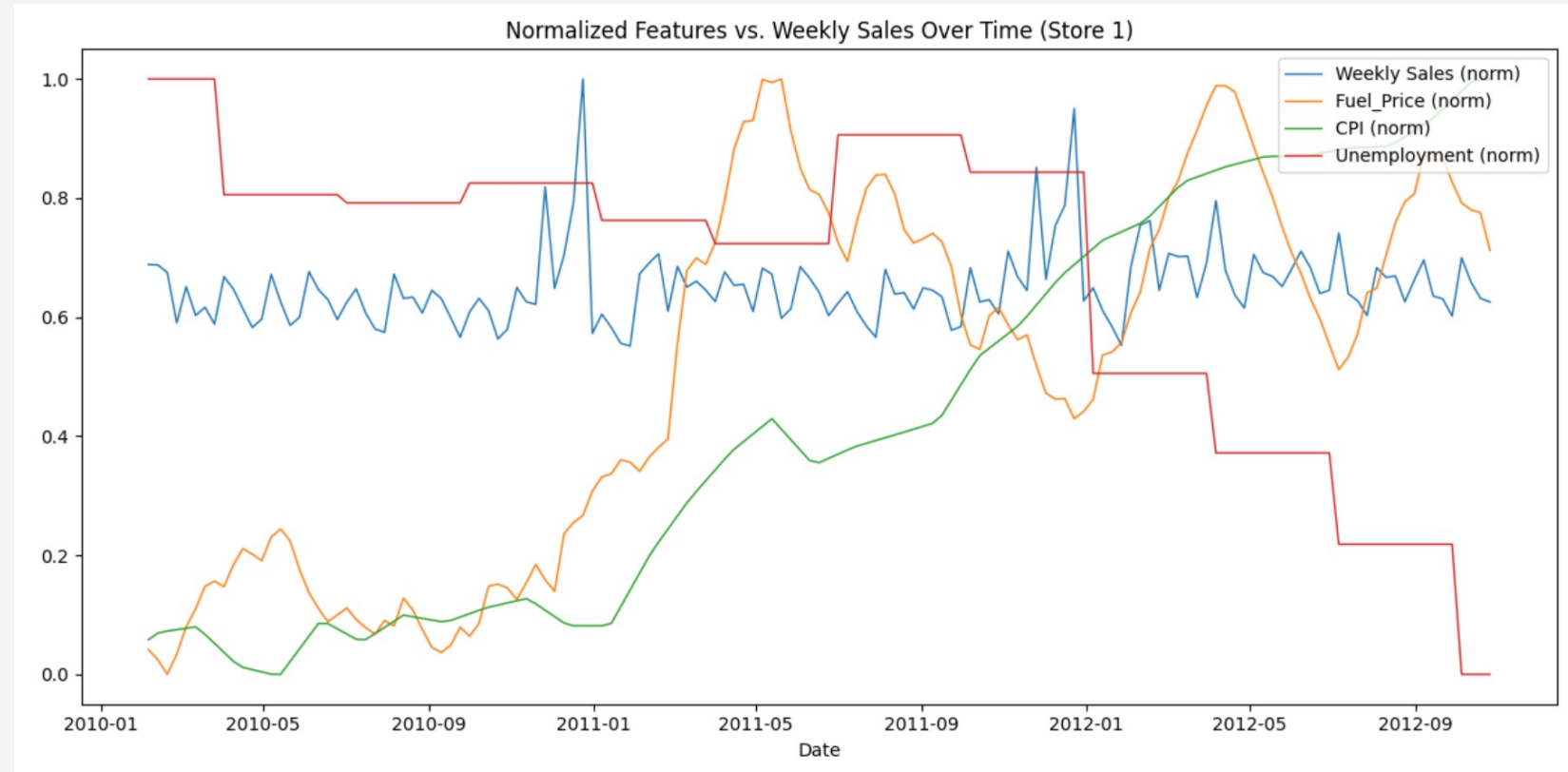
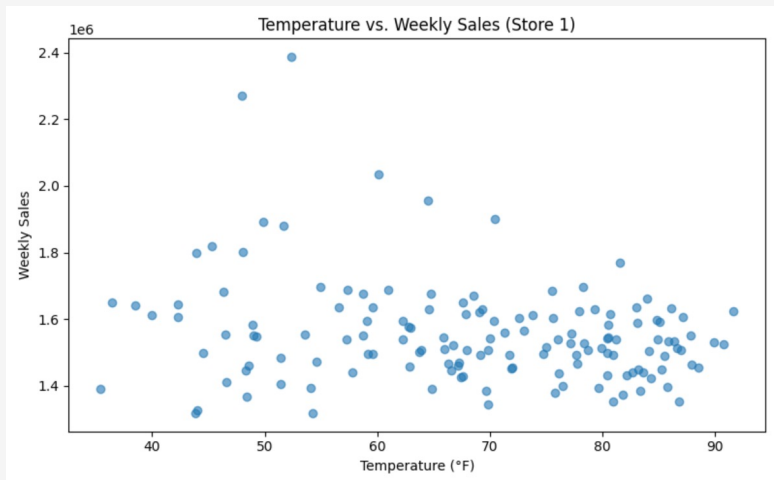
1. 주간 매출 추이



2. Feature별 분포

Temperature - scatter plot

그 외 변수들 - line plot



01

선형회귀분석

여러 독립 변수를 사용해 종속 변수를 예측하는 통계적 모델

각 독립 변수에 대해 가중치(회귀 계수) 학습 후 이 가중치들의 가중합으로 결과 예측



01 학습

▶ 특성별 회귀계수

```
num__Temperature      114636.805266
num__Fuel_Price       129296.238219
num__CPI               10020.492834
num__Unemployment     -43282.856527
num__Month_sin         50198.410886
num__Month_cos        230632.144712
num__Lag_52           849037.521911
cat__Holiday_Flag_1   30995.456815
dtype: float64
```

가장 영향력이 큰 변수: Lag_52 (1년 주기 반영)

→ 연간 반복되는 패턴 강함

Month_cos, Month_sin 도 꽤 강한 영향

→ 계절성 유의미

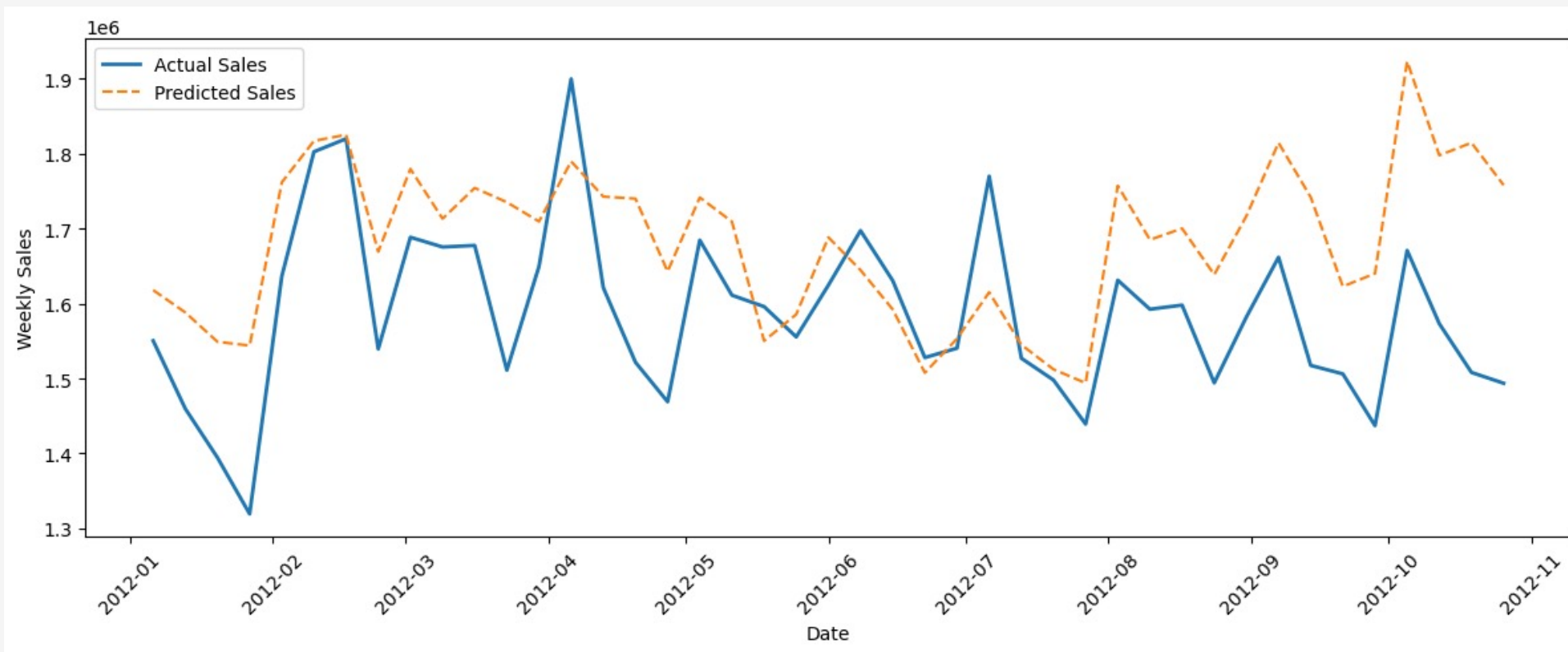
Holiday_Flag_1의 영향도 확인됨

→ 공휴일 특수 수요 존재

Fuel_Price와 Temperature는 양의 상관

02 시각화

▶ 예측 VS 실제 시각화



03 검증

▶ MAE, RMSE, R^2

```
MAE: 115555.72  
RMSE: 139357.12  
 $R^2$ : -0.5013
```

MAE : 예측값과 실제값 사이의 절대적인 차이의 평균

RMSE : 오차 제곱의 평균을 구한 뒤, 그에 루트를 씌운 값

$R^2 < 0$: 단순 평균보다도 예측력이 낮다는 의미

→ 성능 매우 낮음. 과소적합 또는 데이터 특성 반영 부족 가능성.

04 한계 및 보완

원인	구체적 설명	해결 방안
선형 회귀의 한계	비선형적 매출 패턴을 직선으로만 설명하려 하다 보니 과소적합됨	비선형 모델(예: Gradient Boosting, SVM 등)
변수 간 상호작용 <u>미반영</u>	예: CPI + Unemployment 같이 복합적인 경제지표 관계를 고려하지 않음	상호작용 항 추가 혹은 트리 기반 모델 사용
외부 요인 누락	예: 프로모션, 경쟁사, 날씨, 지역 정보 등이 없음	추가 변수 확보 및 모델 구조 강화
시계열 특성 한정적 반영	Lag_52는 좋지만, trend, 추세(linear 증가/감소) 등을 반영하진 못함	trend 피처 추가 (rolling avg, cumulative 등)

02

랜덤포레스트

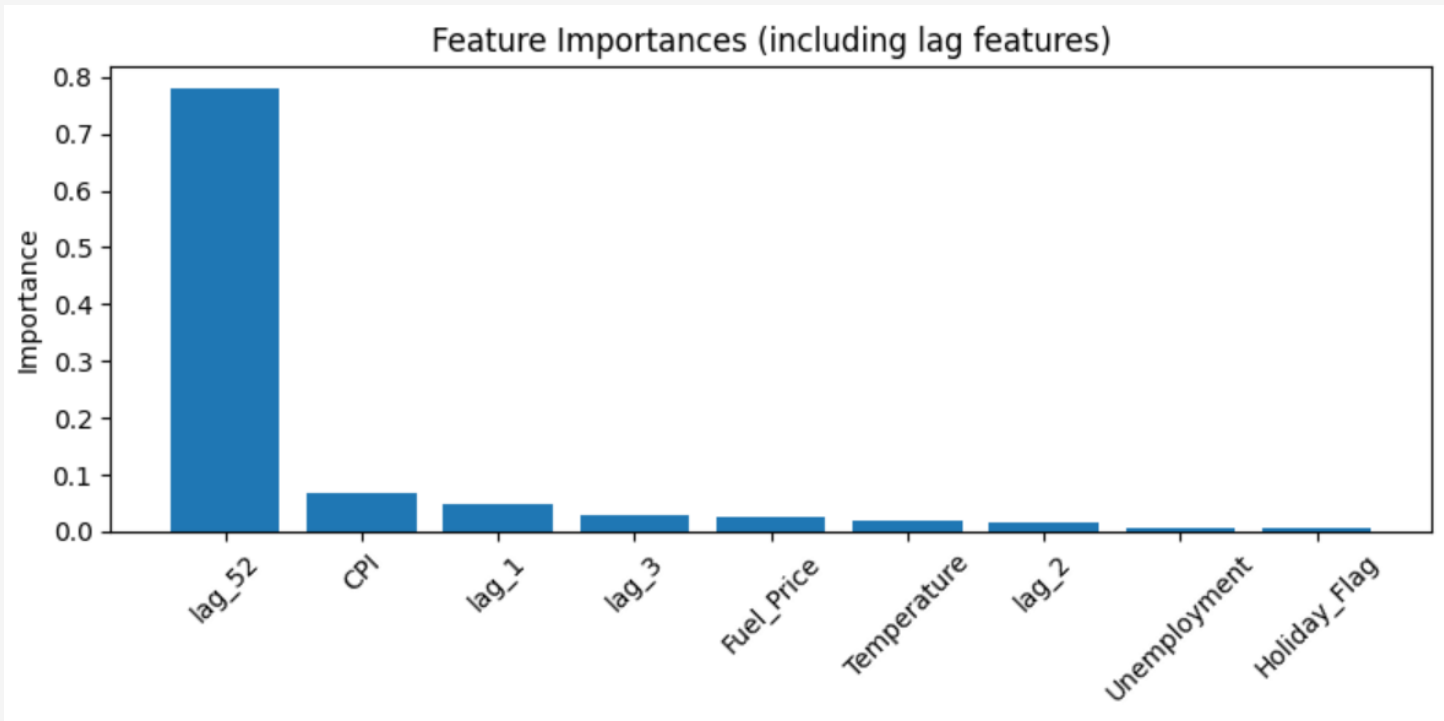
여러 모델을 독립적으로 동시에 학습하는 **배깅(Bagging)** 방식 모델

여러 결정 트리를 **무작위 샘플링**과 **특성 선택**을 통해 각각 훈련 후 예측들의 평균을 결과로 도출

01 학습

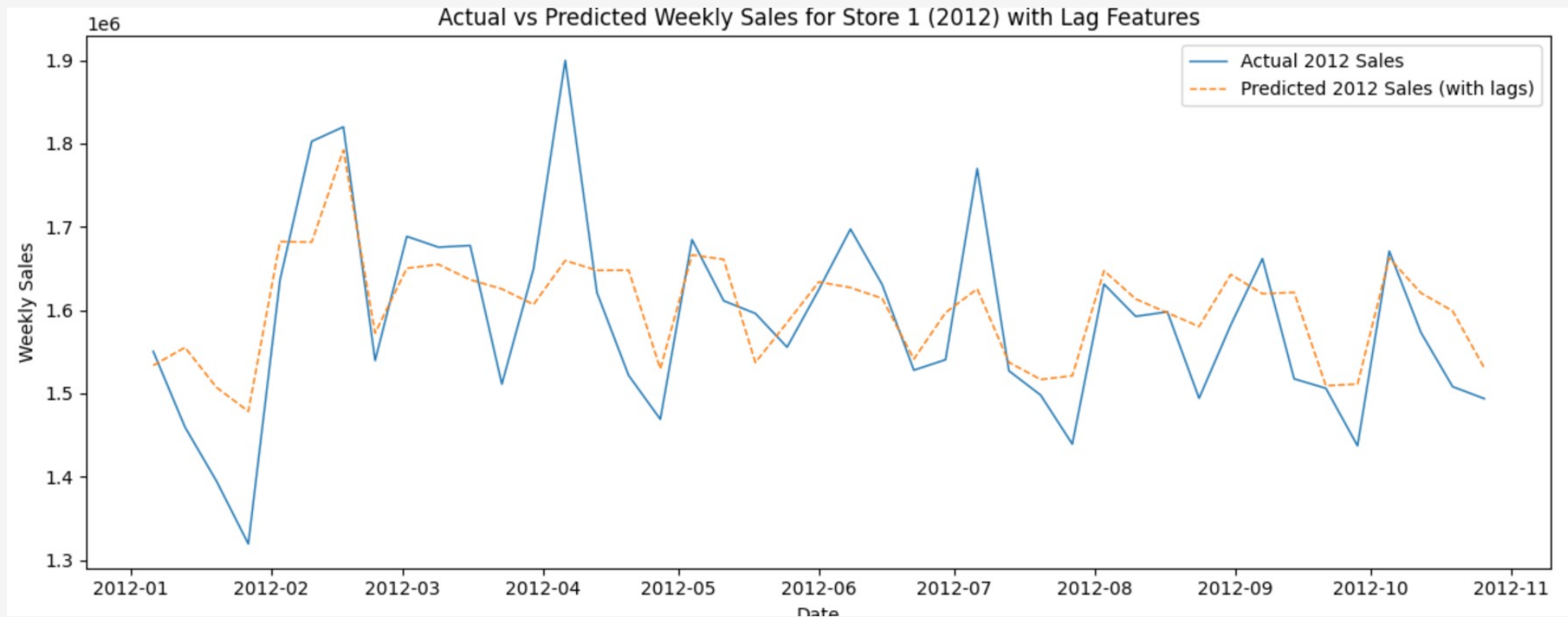
▶ Feature별 중요도

: Lag_52(1년 주기)의 영향이 가장 크게 나타남



02 시각화

▶ 예측 vs 실제



03 검증

▶ RMSE : 75955.62

```
Validation RMSE (2012) with lag features: 75955.62
```

주당 예측 오차가 평균 \$76,000 정도

주간 평균 매출 고려시 약 4.4% 정도 오차

→ **상당히 양호한 수준의 정확도**

04 한계 및 보완

▶ 한계

예측이 일정 주차에서만 크게 빗나감 → 모델 안정성 문제

학습 데이터가 부족 or 특정 시기(예: 연휴, 비수기 등)에 모델이 민감하게 반응하지 못했을 가능성

▶ 보완

잔차(residual) 분석을 통해 어떤 주차에서 오차가 컸는지 확인

예측값 vs 실제값을 시계열 그래프로 시각화해서 패턴 또는 오차 집중 구간 확인

03

XGBoost

eXtreme Gradient Boosting

이전 모델의 오류를 보완하는 방식으로 다음 모델이 학습하는 부스팅(Boosting) 모델

01 학습

▶ 최적 하이퍼파라미터 (RandomizedSearchCV)

```
최적 파라미터: {'subsample': 0.6, 'n_estimators': 150, 'max_depth': 7,  
                'learning_rate': 0.1, 'gamma': 0, 'colsample_bytree': 0.6}
```

subsample, colsample_bytree = 0.6 각 단계에서 60% data, 60% feature 무작위 샘플링

트리 개수 = 150, 최대 깊이 = 7 복잡도 약간 높임

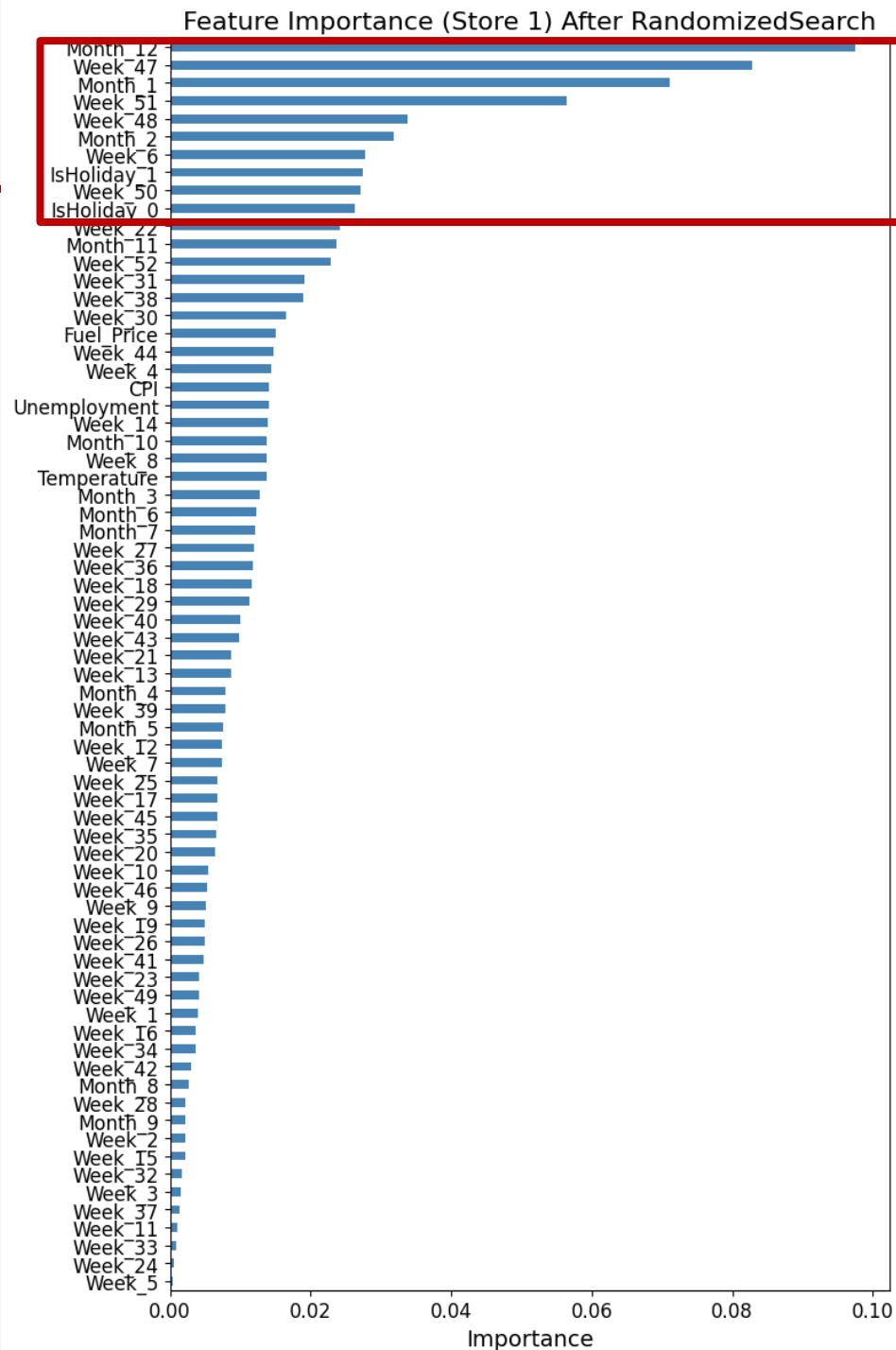
학습률 = 0.1 높은 학습률 사용하여 빠르게 수렴

gamma = 0 리프 노드 분할시 최소 손실 감소 제약 두지 않음

02 시각화

▶ Feature 중요도 Top 10

- | | |
|-----------------------|-------------------|
| 1.Month_12 (12월) | → 크리스마스, 연말 세일 영향 |
| 2.Week_47 (47주차) | → 블랙프라이데이 직전 주간 |
| 3.Month_1 (1월) | → 연초 |
| 4.Week_51 (51주차) | → 크리스마스 후 |
| 5.Week_48 (48주차) | → 블랙프라이데이 |
| 6.Month_2 (2월) | → 연초 |
| 7.Week_6 (6주차) | → 설 연휴 이후 |
| 8.IsHoliday_1 (휴일=1) | → 휴일 유무 |
| 9.Week_50 (50주차) | → 크리스마스 전 |
| 10.IsHoliday_0 (휴일=0) | → 휴일 유무 |



03 검증

▶ RMSE 분석

```
최고 교차검증 RMSE: 98664.70289255939  
검증 세트 RMSE (RandomizedSearch 후): 96073.26
```

최적 교차검증 RMSE VS 실제 검증 세트 RMSE

검증 세트에서 약 2,600달러 정도 더 낮은 RMSE를 기록

이는 데이터 분할 방식(3-폴드 CV vs. 1회 분할 검증)의 차이에서 일부 기인할 수 있음

주간 평균 매출(약 1,600,000달러) 대비 $\approx 6\%$ 내외 오차율

“평균적으로 $\pm \$ 96000$ 정도 어긋날 수 있다”는 의미

04 한계 및 보완

외부 변수 제한

온라인 트래픽, 경쟁사 프로모션 정보, 지역 축제 일정 등은 반영하지 못함

특히 날씨(강수량, 풍속), 미세먼지 지수 등 추가 기후 변수를 넣으면 장기 예측 정확도를 높일 수 있음

하이퍼파라미터 탐색 범위 및 방식

RandomizedSearchCV → $n_iter=20$ 으로 탐색 조합이 제한적

Bayesian Optimization이나 n_iter 확장, GridSearch 세밀 탐색을 병행해 최적화 잠재력을 더 끌어올릴 수 있음

γ 값을 세밀하게 조정하면 트리 분할 기준을 더 정교하게 제어 가능

04

앙상블 학습

여러 개의 모델을 함께 사용해 더 좋은 예측 성능을 내는 학습 방법

단일 모델보다 더 높은 정확도 / 과적합 위험 감소 / 다양한 데이터 특성에 유연하게 대응 가능



01 학습

▶ 앙상블 모델 설명

앞서 모델링한 선형회귀, 랜덤포레스트, XGBoost의 강점들을 살려 스택킹 방식으로 앙상블.

선형회귀: 추세 패턴 포착 우수

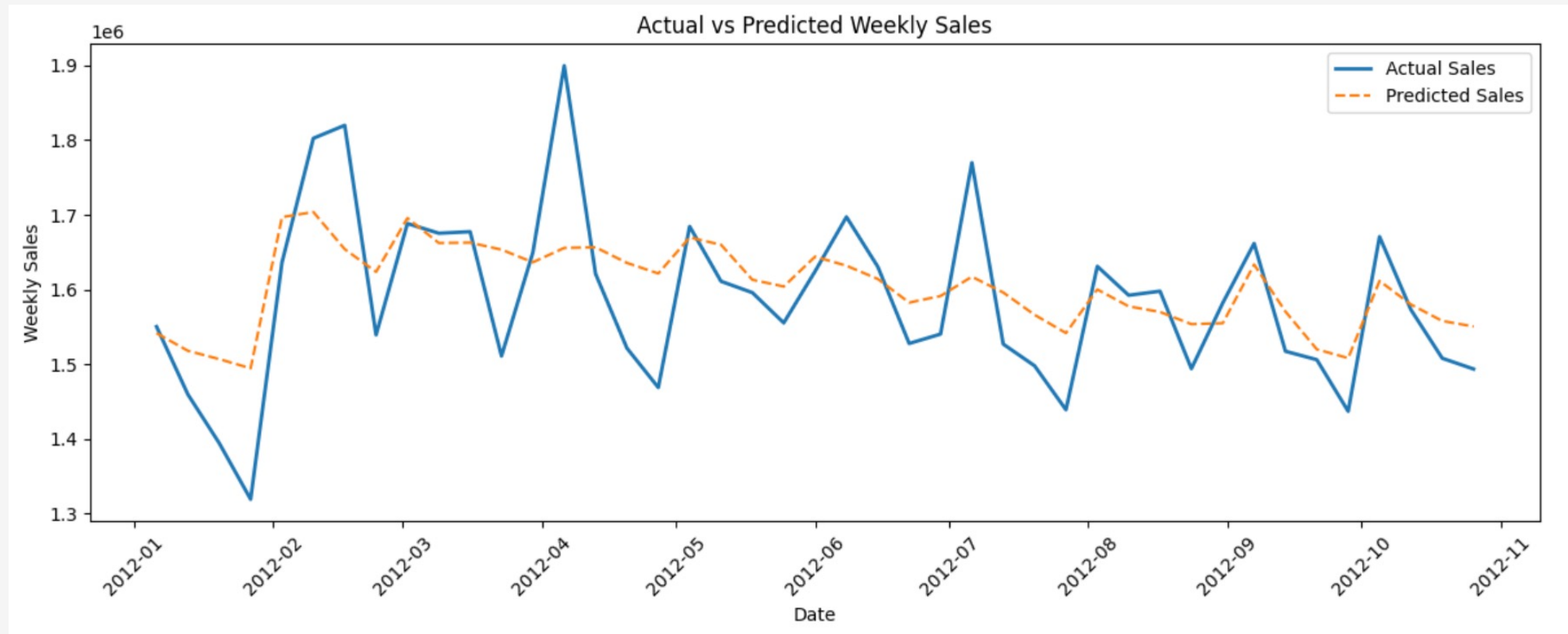
랜덤포레스트: 비선형 관계 처리 강점 발휘

XGBoost: 복잡한 상호작용 모델링 성공

가장 좋은 성능을 갖는 random_state를 찾기 위해 반복하여 학습

02 시각화

▶ 예측 vs 실제



03 검증

```
Best random_state: 48  
Best RMSE: 83011.25  
Corresponding MAE: 63493.59  
Corresponding R2: 0.4673
```

오차 감소

MAE : 선형회귀 대비 약 45% 감소

RMSE : 선형회귀 대비 약 40% 이상 감소

설명력(R²) 증가

R² : -0.5013(음수) → 0.4673(양수)로 크게 상승

데이터의 변동성을 훨씬 잘 설명하며 예측력이 높아졌다는 의미

개별 모델 한계 극복

선형회귀 - 단순 선형 관계만 포착, 랜덤포레스트, XGBoost - 비선형성 잡아냄
이들의 장점을 결합하여, 선형 및 비선형 패턴 모두 반영함으로써 예측력을 높임

04 한계 및 보완

현 모델의 한계점

R^2 0.4673: 전체 변동의 46.73%만 설명

랜덤포레스트 모델보다 RMSE 낮음

잔여 오차 원인: 계절성 패턴 미반영, 외부 충격 요인 미고려

성능 개선을 위한 전략

메타 모델 최적화: RidgeCV 적용 검토

시계열 전용 모델 추가(ARIMA, Prophet)

딥러닝 기반 LSTM 네트워크 통합

결과 분석

모델명	RMSE
선형회귀분석	139357.12
랜덤포레스트	75955.62
XGBoost	96073.26
앙상블학습	83011.25

RMSE가 가장 작은 **랜덤포레스트** 모델의 성능이 가장 좋음

주당 예측 오차가 평균 \$76,000 정도

주간 평균 매출 고려시 약 4.4% 정도 오차

* RMSE 선택 이유 : 모든 모델에서 공통적으로 평가 지표로 사용

한계 및 개선 방안

1. 외부 요인 반영 부족

경쟁사 프로모션 정보, 축제 일정(예. 올림픽), 날씨(예. 강수량, 미세먼지) 등은 반영하지 못함.

이벤트 일정, 기상 정보, 경쟁사 동향 등을 수집·정제하여 변수로 포함하면 예측 정확도를 높일 수 있을 것으로 기대

2. 데이터 크기

연간 주기를 반영하기에 다소 작은 데이터. 이는 계절성이나 장기적인 추세를 학습하는 데 한계.

충분한 크기의 데이터를 확보한다면 모델의 예측 정확도를 더 높일 수 있을 것으로 기대.

3. 주간 데이터

일일 매출 데이터를 구하기 어려워 주간 매출 데이터를 사용하였으나, 일일 단위에서 발생하는 세부적인 변화(예: 주중/주말 간 매출 차이, 특정 요일의 날씨 영향 등)를 반영하기 어려웠음. 일일 매출 데이터가 확보된다면 보다 정밀한 반영 가능할 것.



Thank you

QR코드를 스캔하시면 프로젝트 코드와 관련 문서를 모아둔
Notion 페이지로 이동합니다

