[유통데이터 활용 경진대회 수요예측 부문 분석보고서 서식]

제3회 유통데이터 활용 경진대회 (수요예측부문) 분석보고서

1. 팀명 : FC BOAZ

2. 세부내용

◦ 분석 개요

1. 분석 목적

본 보고서는 연관분석(Association Analysis) 기법을 활용하여 유통 데이터에서 1데이터의 중분류 '라면, 통조림, 상온즉석' 제품군 및 2데이터의 대분류 '면류, 라면류' 제품군의 수요를 예측하는 것을 목적으로 함. 이 분석의 주된 목표는 도매점에서 함께 발주되는 제품들 간의 상관관계를 주차 별로 파악하여, 특정 제품이 주문될 때 다른 제품의 동반 주문 가능성을 예측하는 것임. 이를 통해 제품 간의 상호 연관성을 바탕으로 수요 패턴을 분석하고, 향후 발주될 가능성이 높은 제품 조합을 예측함으로써 더 정교한 수요 예측을 수행하고자 함.

연관분석을 통해 도출된 인사이트는 예로 '라면, 통조림, 상온즉석'이 독립적으로 소비되는 것이 아니라 상호 연관성을 가질 수 있음을 보여주며, 이를 기반으로 도매점이 재고 관리 및 공급 계획을 더욱 효율적으로 수립할 수 있음. 특히, 주요 제품군의 동반 수요를 예측함으로써 재고 부족 및 과잉을 방지하고, 공급망 운영의 효율성을 높이는 데 기여할 수 있음. 궁극적으로, 본 분석은 도매점의 전체적인 판매 흐름을 이해하고, 발주 패턴에 따른 수요 변화에 신속하게 대응할 수 있도록 지원하며, 유통망의 전반적인 효율성을 강화하는 데 목적이 있음.

2. 데이터 분석 EDA

Python 라이브러리 Sweetviz를 활용하여 '판매수량'을 타겟 변수로 지정한 보고서 파일을 생성하고 데이터를 탐색함. 또한, Matplotlib을 이용한 시각화와 SQLite3를 활용한 데이터 핸들링을 통해 탐색적 데이터 분석(EDA)을 진행함

[1데이터]

① 변수별 타겟변수 '판매수량'과의 상관계수

NUMERICAL ASSOCIATIONS
(PEARSON, -1 to 1)

입수	-0.02
매출처코드	-0.02
우편번호	0.01
상품 바코드(대한상의)	-0.00

CATEGORICAL ASSOCIATIONS
(CORRELATION RATIO, 0 to 1)

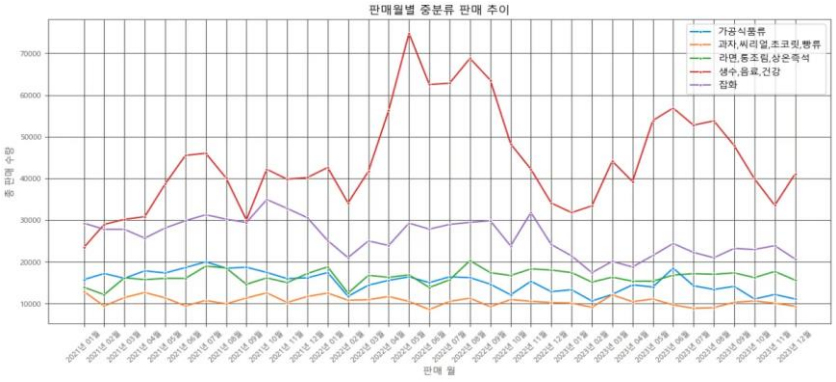
옵션코드	0.15
중분류	0.15
소분류	0.10
대분류	0.03
구분	0.01

수치형 변수 중에는 입수, 매출처코드, 우편번호가 비수치형 변수 중에는 옵션코드, 중분류, 대분류가 '판매수량'과 높은 상관관계를 갖는 것으로 나타남

② 판매수량 월별 판매 및 발주 추이

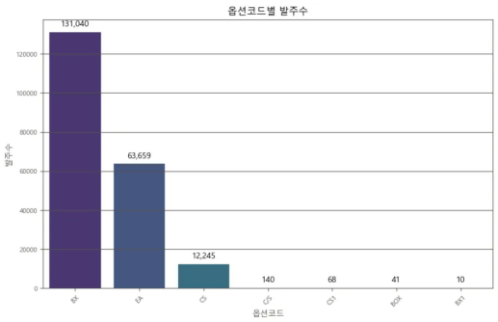
전반적으로 매년 11월부터 1월까지 판매 및 발주량이 감소하고, 이후 급격한 증가가 나타나는 것으로 보아 주기적인 변동성이 있는 것으로 판단됨

③ 판매수량 상위 5개 중분류 월별 판매 추이



'생수, 음료, 건강'이 지속적으로 가장 높은 판매수량이 나타나며, 특히 해당 중분류의 경우 2022년 판매수량이 급증함. '라면, 통조림, 상온즉석'의 경우 비교적 평이한 추세를 보이고 있어 계절적 영향을 크게 받지 않는 것으로 보임.

④ ‘라면, 통조림, 상온즉석’ 옵션코드별 발주수



박스 단위가 압도적으로 많은 발주수가 나타났으며, 이어서 최소 단위와 묶음 단위가 많은 발주수가 집계됨.

[2데이터]

① 변수별 타겟변수 ‘판매수량’과의 상관계수

NUMERICAL ASSOCIATIONS
(PEARSON, -1 to 1)

입수	-0.11
상품 바코드(대한상의)	-0.08
우편번호	0.02
매출처코드	-0.00

CATEGORICAL ASSOCIATIONS
(CORRELATION RATIO, 0 to 1)

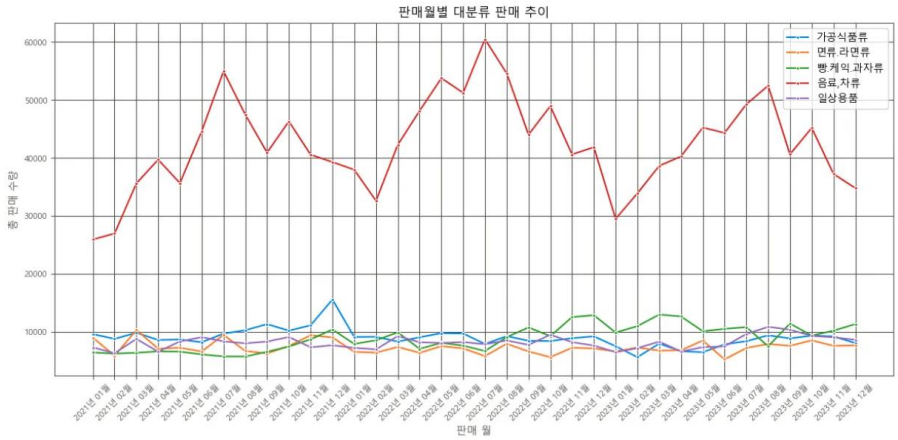
중분류	0.20
대분류	0.18
옵션코드	0.14
구분	0.00

수치형 변수 중에는 입수, 상품 바코드, 우편번호가 비수치형 변수 중에는 중분류, 대분류, 옵션코드가 ‘판매수량’과 높은 상관관계를 갖는 것으로 나타남

② 판매수량 월별 판매 및 발주 추이

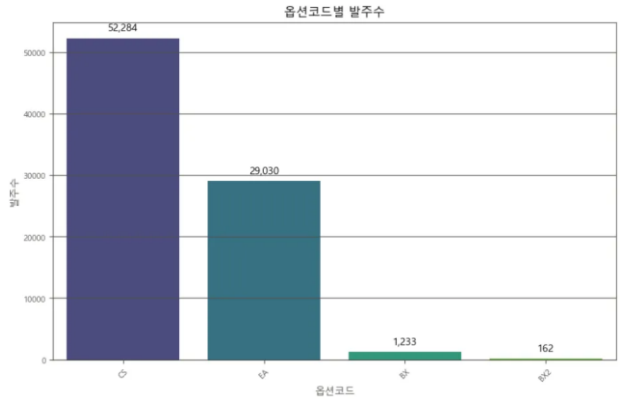
전반적으로 매년 11월부터 1월까지 판매 및 발주량이 감소하고, 이후 급격한 증가가 나타나는 것으로 보아 주기적인 변동성이 있는 것으로 판단됨

③ 판매수량 상위 5개 중분류 월별 판매 추이



'음료, 차류'가 지속적으로 가장 높은 판매수량이 나타남. '라면, 통조림, 상온즉석' 의 경우 비교적 평이한 추세를 보이고 있어 계절적 영향을 크게 받지 않는 것으로 보임.

④ '면류, 라면류' 옵션코드별 발주수



묶음 단위가 가장 많은 발주수가 나타났으며, 이어서 최소 단위와 박스 단위가 많은 발주수가 집계됨.

3. 데이터 전처리

[반품 데이터 처리]

판매수량을 예측해야 하므로 '구분' 컬럼에서 '반품' 값을 보유한 행은 별도의 처리가 필요하다고 판단함. '매출' 값의 경우 판매된 행이므로 양수로 표시되고, '반품' 값의 경우 음수로 표시됨. 따라서 같은 판매일이면서 같은 상품 바코드끼리 ('매출' 행의 판매수량)+('반품' 행의 판매수량)으로 계산함. 합산을 한 이유는 '반품' 행 판매수량의 경우 이미 음수 값으로 표시되어 있기 때문에 합산으로 처리함. 1데이터와 2데이터 모두 동일하게 적용함.

[판매일 변수 처리]

2데이터의 경우 1데이터와 다르게 판매일 변수가 다르게 구성됨. 1데이터의 경우 2021년 1월부터 2023년 12월까지 모든 날짜가 기록되어 있지만, 2데이터의 경우 2021년 1월부터 2023년 5월까지 매달 말일만 기록되고, 2023년 6월부터는 모든 날짜가 기록되어 있음.

따라서 아래 예시를 기반으로 2데이터 판매일 변수를 1데이터와 동일하게 재구성함.

1월 31일(매달 말일)로 되어있는 행 약 2100개를 1월1일부터 31일까지로 각각 할당. 예를 들어 1월1일 70개, 1월2일 70개, 1월 3일 70개, 1월 4일 70개 등. 이렇게 할당해서 2023년 5월까지 월별로 되어있던 판매일을 일별로 재구성함.

[이상치 데이터 처리]

판매수량 값의 이상치를 탐지하고 제거하는 과정을 통해 시계열 데이터의 신뢰도를 높이려 함. 판매일 데이터를 주 단위로 재구성한 후, 옵션코드 변수의 값별(BX, CS, EA)로 데이터를 분할하여 각각에 대해 이상치 탐지를 수행함. 각 단위의 판매수량 집계 방식이 다르기 때문에 데이터 분할이 필요했으며, 탐지 방법으로는 IQR 사분위, Standard Deviation Method, Z-score Method, Isolation Forest 네 가지 방식을 사용함.

① IQR 사분위(IQR: Interquartile Range)

IQR 방법은 1사분위(Q1)와 3사분위(Q3)의 차이를 기반으로 이상치를 탐지함. Q1보다 1.5배 IQR 이하이거나, Q3보다 1.5배 IQR 이상인 값들이 이상치로 간주됨. 이는 극단값에 영향을 덜 받는 방식으로, 주로 데이터가 비대칭인 경우에도 안정적인 결과를 제공함.

② 표준편차 방법(Standard Deviation Method)

표준편차 방법은 평균으로부터 일정 범위를 벗어난 값들을 이상치로 간주함. 평균에서 k배의 표준편차 범위 밖에 있는 값들이 이상치로 간주되며, k 값은 보통 2 또는 3으로 설정합니다. 이 방법은 데이터가 정규분포를 따를 때 효과적임.

③ Z-score 방법

Z-score는 데이터 값이 평균에서 얼마나 떨어져 있는지를 표준편차 단위로 나타내는 방법임. 일반적으로 Z-score가 3 이상인 값들은 이상치로 간주됩니다. 이 방법 역시 데이터가 정규분포를 따를 때 유효함.

④ Isolation Forest

Isolation Forest는 비지도 학습 알고리즘으로, 데이터를 여러 이진 트리로 나누며 이상치를 탐지함. 일반적인 데이터보다 이상치는 더 적은 분할을 통해 고립되기 때문에, 이 고립도를 기준으로 이상치를 판단함. 다른 방법들과 달리, 데이터 분포에 관계없이

사용할 수 있으며 복잡한 패턴을 감지하는 데 유리함.

이상치를 탐지할 때, 단일 방법에 의존할 경우 특정 방법의 한계로 인해 잘못된 이상치가 포함될 수 있음. 예를 들어, Z-score와 같은 방법은 정규분포 가정을 전제로 하므로, 데이터가 정규분포를 따르지 않을 경우 이상치가 과대 탐지될 수 있음. 반면, Isolation Forest는 비지도 학습 알고리즘으로 다양한 데이터 패턴을 감지할 수 있지만, 트리의 깊이에 따라 결과가 변동될 수 있음.

따라서, IQR, Standard Deviation, Z-score, Isolation Forest 네 가지 방법에서 탐지된 이상치 중 공통적인 이상치만 필터링하는 방법을 채택하여 각 방법의 한계를 보완함. 모든 방법에서 일관되게 탐지된 데이터만을 이상치로 간주함으로써 보다 신뢰성 높은 이상치 탐지 결과를 도출함.

4. 변수 추가

[공공데이터]

판매 데이터로 이해할 수 없는 시장의 외부 요인에 대한 정보를 통해 정확하고 신뢰성 높은 예측을 하기 위해 공공데이터를 사용하였음. 실업률에 대하여 주어진 데이터 상에서 성별에 따른 소비 패턴을 확인할 수 없었지만, 경제적 요인의 영향을 포괄적으로 판단하기 위해 전체 실업률과 성별 실업률을 함께 고려하였음. 또한, GDP는 분기별 값을 월별로 적용하였으며, 그 외 모든 데이터는 2021년부터 2023년까지 월에 해당되는 값으로 이루어져 있음. 공공데이터로 소비자물가지수, 생활물가지수, 실업률(전체, 남성, 여성), GDP(명목, 실질), 소비자심리지수, 환율, 기준금리를 활용하였으며, 각 데이터의 출처는 하단 활용데이터 난에 첨부하였음.

[매출처코드별 평균 발주 기간 및 총 발주 건수 컬럼 추가]

각 매출처코드별 '평균 발주 기간'과 '총 발주 건수'를 산출하였음. 이는 각 거래처의 발주 패턴을 이해하고, 수요 예측 모델을 개선하기 위한 목적이 있음. 평균 발주 기간은 각 매출처가 발주를 진행하는 일자 간의 간격을 의미하며, 이를 통해 주기성을 확인할 수 있음. 총 발주 건수는 전체 기간 동안 거래처의 발주 건수의 합계를 의미하며, 각 매출처의 활동 수준과 규모를 직관적으로 파악할 수 있음.

◦ 분석 방법 및 절차

[연관분석 개요]

매출처코드와 대분류별로 판매 수량을 집계하여 장바구니 형태의 데이터프레임을 만듦. 그리고 장바구니 데이터프레임의 값을 1(판매됨) 또는 0(판매되지 않음)으로 이진화 시킨 후, Apriori 알고리즘을 사용하여 빈번한 상품 집합, 즉 구매할 때 같이 구매되는 경향이 있는 제품들을 찾아줌.

신뢰도(confidence)의 최소 임계값을 0.5로 설정하여 해당하는 값들만 따로 확인하였음.

```
# 매출처코드별로 대분류를 one-hot encoding하여 장바구니 행렬 생성
basket = (date_data
          .groupby(['매출처코드', '대분류'])['판매일']
          .count()
          .unstack()
          .reset_index()
          .fillna(0)
          .set_index('매출처코드'))

# 1과 0으로 이진화
basket_sets = basket.applymap(lambda x: 1 if x > 0 else 0)

# Apriori 알고리즘을 사용하여 빈번한 아이템 집합 찾기 (지지도 최소값 0.01)
frequent_itemsets = apriori(basket_sets, min_support=0.01, use_colnames=True)

# 연관 규칙 생성
rules = association_rules(frequent_itemsets, metric="confidence", min_threshold

# Antecedents에 '라면, 통조림, 상온즉석' 이 포함된 규칙만 필터링
rules_filtered = rules[rules['antecedents'].apply(lambda x: '면류.라면류' in x)]
```

단일 상품간의 연관 규칙만을 확인하기 위해서 여러 개의 제품이 묶여 있는 컬럼은 제거함. 여러 개의 제품 간의 연관은 제거하여 단일 제품 관계만 확인

변경 전		변경 후	
frozenset({'가공식품류', '조미료류', '일상용품', '통조림류'})	frozenset({'면류.라면류', '빵.케익.과자류', '음료,차류'})	일상용품	음료,차류
frozenset({'빵.케익.과자류', '조미료류', '일상용품', '통조림류'})	frozenset({'면류.라면류', '가공식품류', '음료,차류'})	통조림류	음료,차류
		조미료류	일상용품

앞서 만든 데이터프레임을 활용하여 매출처 코드 별로 '평균발주일자'와 '총발주건수'를 만들어주고, merge 함수를 사용하여 장바구니 분석 결과를 매출처코드별 붙여 주었음.

이후 1데이터에서는 '중분류'값이 '라면, 통조림, 상온즉석'인 데이터만, 2데이터에서는 '대분류'값이 '면류, 라면류'인 값들만 남겨 주었음.

[모델링 진행 개요]

시계열 모델인 Prophet과 SARIMA, 그리고 딥러닝 계열 모델인 LSTM과 GRU를 활용하여 판매량 예측을 수행함

◆ 데이터 준비

- 장바구니 분석 컬럼이 결합된 전처리 데이터에서 불필요한 컬럼 삭제함
  <삭제 컬럼>
  - 원본 데이터에서 '판매일', '판매수량' 제외한 모든 컬럼
  - 전체 데이터가 FALSE인 컬럼
- '판매일' 주 단위를 기준으로 데이터를 그룹화하여 판매수량은 합계로, 다른 수치형 컬럼은 평균으로 산출함
- 집계된 데이터에 2024년 데이터 추가
  - 공공데이터 이외의 데이터는 2023년도 데이터 사용
- 판매수량과의 상관계수 절댓값이 0.1 이상인 독립변수만 남기고, 나머지 변수는 삭제. 이 과정을 통해 두 개의 데이터셋(삭제된 데이터셋과 삭제되지 않은 데이터셋)을 생성하고, 각각 모델링을 진행하여 성능을 비교

[모델 구현 과정]

1. Prophet

A. 데이터 준비 및 훈련/테스트 데이터 분할

i. 모델 학습을 위해 열 이름을 ds 와 y 로 변경

ii. 첫 36 개 행을 훈련 데이터로, 24 년도 데이터인 마지막 26 개 행을 테스트 데이터로 분할함

iii. '판매수량' 컬럼을 제외한 설명변수를 선택하여 훈련 및 테스트 데이터셋을 생성함

iv. 한국 공휴일 데이터 프레임 생성

ㄱ. Prophet 모델에 통합하여 예측 성능을 향상시키기 위함

ㄴ. 유통업 특성상 제품은 재고로 쌓아두고 판매되기 때문에, 공휴일이 미치는 영향은 장기간 지속될 것으로 예상함. 이에 따라, 공휴일 기준으로 30 일 이전과 이후의 기간에도 영향을 미치는 것으로 설정함

ㄷ. 데이터 프레임 예시

holiday	ds	lower_window	upper_window
새해	2021-01-01	-30	30
설날	2021-02-11	-30	30

설날	2021-02-12	-30	30
부처님오신날	2021-05-19	-30	30

B. 하이퍼파라미터 튜닝

i. 모델의 성능을 최적화하기 위해 다양한 하이퍼파라미터 조합을 설정

ㄱ. changepoint_prior_scale 에 대해 4 개의 값, seasonality_prior_scale 에 대해 3 개의 값, seasonality_mode 에 대해 2 개의 값을 설정함

ii. 하이퍼파라미터의 모든 조합을 생성하기 위해 itertools.product 를 사용하여, 각 하이퍼파라미터의 모든 가능한 조합을 리스트로 조합을 설정함

```
# 하이퍼파라미터 조합 설정
param_grid = {
    'changepoint_prior_scale': [0.01, 0.05, 0.2, 0.3],
    'seasonality_prior_scale': [1.0, 5.0, 10.0],
    'seasonality_mode': ['additive', 'multiplicative'],
}
```

C. 모델 학습 및 평가

i. 각 조합에 대해 Prophet 모델을 초기화하고, 모델을 학습하고 평가함

```
for changepoint_prior, seasonality_prior, seasonality_mode in param_combinations:
    # Prophet 모델 적합
    model = Prophet(
        changepoint_prior_scale=changepoint_prior,
        seasonality_prior_scale=seasonality_prior,
        seasonality_mode=seasonality_mode,
        yearly_seasonality=True,
        weekly_seasonality=True,
        daily_seasonality=False,
        holidays=holidays_corrected
    )
```

- changepoint_prior_scale: 변곡점 감지의 민감도를 조절함
- seasonality_prior_scale: 계절성의 영향을 조절함
- seasonality_mode: 계절성을 추가하는 방식으로, 'additive' 또는 'multiplicative'를 선택함
- yearly_seasonality, weekly_seasonality, daily_seasonality: 계절성을 설정하는 파라미터
- holidays: 공휴일 정보를 포함하여 모델의 예측 성능을 개선함

ii. 모델을 훈련 데이터로 학습시키고 미래 데이터 프레임을 생성하고 예측함

```python
# 모델 학습
model.fit(train_df)

# 미래 데이터프레임 생성 (N개 행)
future = model.make_future_dataframe(periods=26, freq='W')
# 예측 수행
forecast = model.predict(future)

# 예측 결과 추출
y_true = test_df['y'].values  # 테스트 데이터의 실제 값
y_pred = forecast['yhat'].values[-26:]
```

iii. 현재의 MAE 가 이전 최적 MAE 보다 낮은 경우, 최적의 하이퍼파라미터와 예측 결과를 업데이트하며, MAE 와 RMSE 를 사용하여 모델 성능을 측정하여, 최적의 하이퍼파라미터 조합과 해당 모델의 성능 지표 선정함

2. SARIMA

A. 데이터 준비 및 훈련/테스트 데이터 분할

i. 첫 36개 행을 훈련 데이터로, 24년도 데이터인 마지막 26개 행을 테스트 데이터로 분할함

ii. '판매수량' 컬럼을 제외한 설명변수를 선택하여 훈련 및 테스트 데이터셋을 생성함

B. 최적의 SARIMA 모델 탐색

i. SARIMA 모델을 자동으로 최적화하기 위해 auto_arima 함수를 사용하며, 계절성을 반영할 수 있도록 seasonal=True를 설정하고, 계절 주기를 월 단위로 설정함

```python
# 최적의 SARIMA 모델 찾기
sarima_model = auto_arima(
    train['sales'],
    exogenous=exogenous_train,
    seasonal=True,
    m=12,  # 월 단위 계절 주기
    start_p=0, start_q=0,
    max_p=5, max_q=5,
    start_P=0, start_Q=0,
    max_P=5, max_Q=5,
    d=None, D=None,
    seasonal_test='ocsb',
    trace=True,
    error_action='ignore',
    suppress_warnings=True,
    stepwise=True,
    n_jobs=1
)
```

- trace=True: 모델 학습 과정에서의 진행 상황을 출력함
- error_action='ignore': 오류 발생 시 무시하도록 설정함
- stepwise=True: 단계별로 모델을 구축하여 최적의 파라미터를 찾음

C. 모델 학습 및 평가

i. 최적의 SARIMA 모델을 사용하여 향후 26주에 대한 예측을 수행함

ii. 예측 결과를 테스트 데이터와 비교하여 성능을 평가하며, MAE(Mean Absolute Error)와 RMSE(Root Mean Squared Error)를 계산하여 모델의 예측 정확도를 측정함

3. LSTM

A. 데이터 준비 및 훈련/테스트 데이터 분할

i. 시계열 데이터 처리를 위해 '판매일' 컬럼을 인덱스로 지정

ii. 종속변수인 '판매수량' 컬럼을 제외한 독립변수 MinMax 스케일링 진행

iii. 딥러닝 모델 특성상 시계열 데이터 예측을 위한 4주 단위 시퀀스 데이터 생성

```
# 시계열 데이터 예측을 위한 생성 함수
def create_sequences(X, y, time_steps=4):
    Xs, ys = [], []
    for i in range(len(X) - time_steps):
        Xs.append(X[i:i + time_steps])
        ys.append(y[i + time_steps])
    return np.array(Xs), np.array(ys)

# 시퀀스 데이터 생성 (4주 단위)
time_steps = 4
X_seq, y_seq = create_sequences(X_scaled, y.values, time_steps)
```

iv. 2024년 1월 이전 데이터는 학습, 2024년 1월부터 6월 데이터는 테스트 데이터로 분할

B. 모델 학습 및 평가

i. 모델 생성 및 성능 개선을 위한 최적의 파라미터 적용

ㄱ. 모델 함수 생성

```
# LSTM 모델 구축
def build_lstm_model(units=50):
    model = Sequential()
    model.add(LSTM(units, activation='relu', input_shape=(X_train.shape[1], X_train.shape[2])))
    model.add(Dense(1))
    model.compile(optimizer='adam', loss='mse')
    return model
```

ㄴ. epochs=50, batch_size=16, verbose=0 로 파라미터 적용

ㄷ. 모델을 훈련 데이터로 학습

```
def train_and_evaluate_model(model, X_train, y_train, X_test, y_test):
    model.fit(X_train, y_train, epochs=50, batch_size=16, verbose=0)
    y_pred = model.predict(X_test)
    rmse = np.sqrt(mean_squared_error(y_test, y_pred))
    mae = mean_absolute_error(y_test, y_pred)
    return y_pred, rmse, mae

# LSTM 모델 학습 및 평가
lstm_model = build_lstm_model()
y_pred_lstm, rmse_lstm, mae_lstm = train_and_evaluate_model(lstm_model, X_train, y_train, X_test, y_test)
```

4. GRU

A. 데이터 준비 및 훈련/테스트 데이터 분할

i. 시계열 데이터 처리를 위해 '판매일' 컬럼을 인덱스로 지정

ii. 종속변수인 '판매수량' 컬럼을 제외한 독립변수 MinMax 스케일링 진행

iii. 딥러닝 모델 특성상 시계열 데이터 예측을 위한 4주 단위 시퀀스 데이터 생성

```
# 시계열 데이터 예측을 위한 생성 함수
def create_sequences(X, y, time_steps=4):
    Xs, ys = [], []
    for i in range(len(X) - time_steps):
        Xs.append(X[i:i + time_steps])
        ys.append(y[i + time_steps])
    return np.array(Xs), np.array(ys)

# 시퀀스 데이터 생성 (4주 단위)
time_steps = 4
X_seq, y_seq = create_sequences(X_scaled, y.values, time_steps)
```

iv. 2024년 1월 이전 데이터는 학습, 2024년 1월부터 6월 데이터는 테스트 데이터로 분할

B. 모델 학습 및 평가

i. 모델 생성 및 성능 개선을 위한 최적의 파라미터 적용

ㄱ. 모델 함수 생성

```
# GRU 모델 구축
def build_gru_model(units=50):
    model = Sequential()
    model.add(GRU(units, activation='relu', input_shape=(X_train.shape[1], X_train.shape[2])))
    model.add(Dense(1))
    model.compile(optimizer='adam', loss='mse')
    return model
```

ㄴ. epochs=50, batch_size=16, verbose=0로 파라미터 적용

```
def train_and_evaluate_model(model, X_train, y_train, X_test, y_test):
    model.fit(X_train, y_train, epochs=50, batch_size=16, verbose=0)
    y_pred = model.predict(X_test)
    rmse = np.sqrt(mean_squared_error(y_test, y_pred))
    mae = mean_absolute_error(y_test, y_pred)
    return y_pred, rmse, mae

# LSTM 모델 학습 및 평가
lstm_model = build_lstm_model()
y_pred_lstm, rmse_lstm, mae_lstm = train_and_evaluate_model(lstm_model, X_train, y_train, X_test, y_test)
```

ㄷ. 모델을 훈련 데이터로 학습

◦ 분석 결과

1. Prophet

D. 최적의 하이퍼파라미터 및 예측 결과

<1데이터>

- 상관분석을 진행한 데이터셋과 진행하지 않은 결과가 동일함

① 최적의 하이퍼파라미터

- changepoint_prior_scale: 0.01
- seasonality_prior_scale: 5.0
- seasonality_mode: 'multiplicative'

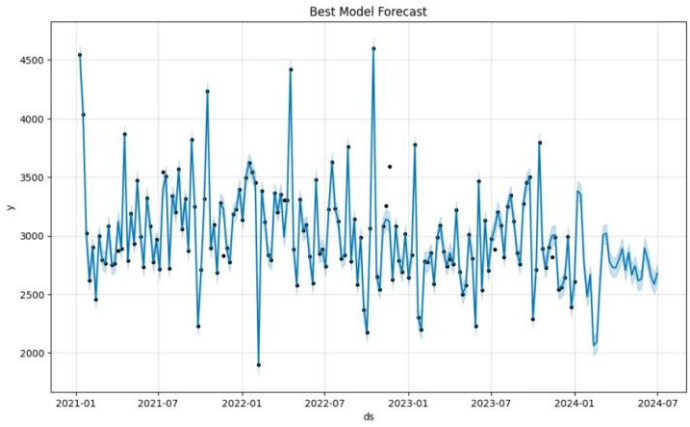
② 성능지표

MAE: 267.3775352975817

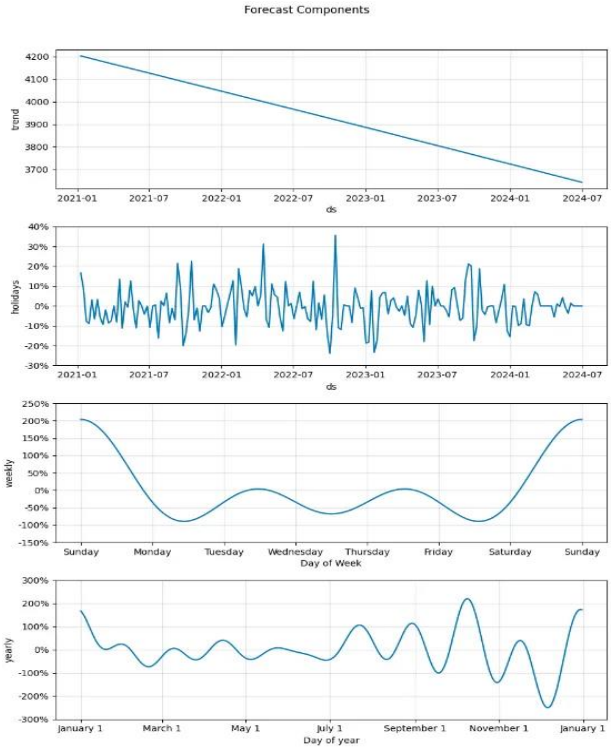
RMSE: 345.62705989119394

③ 예측결과

Prophet 최적 모델 예측 결과 시각화 ↓



Prophet 최적 모델 예측 구성 요소 시각화 ↓



연월	예측값
2024-01	11978.856735
2024-02	9429.594868
2024-03	14262.557779
2024-04	11250.246440
2024-05	10654.244656
2024-06	13606.050073

<2데이터>

- 상관분석을 진행한 데이터 셋

① 최적의 하이퍼 파라미터

- changepoint_prior_scale: 0.01
- seasonality_prior_scale: 5.0
- seasonality_mode: 'multiplicative'

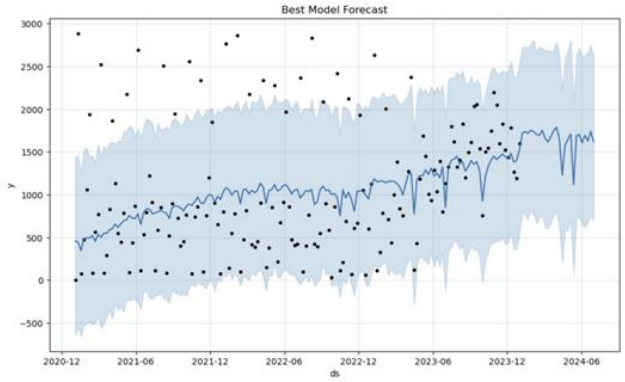
② 성능지표

MAE: 710.9824727034639

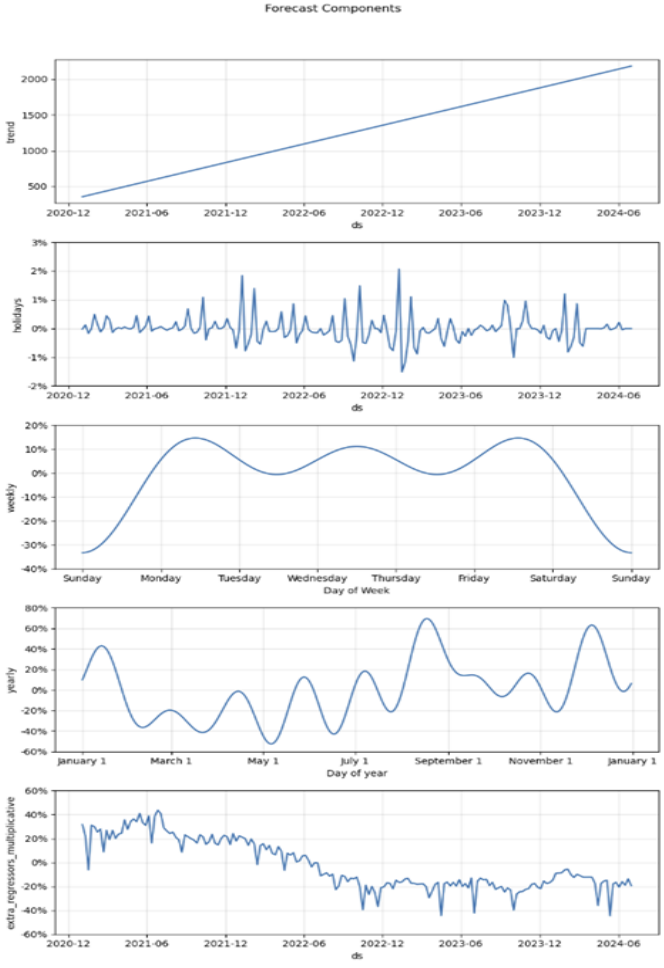
RMSE: 837.9476429563201

③ 예측결과

Prophet 최적 모델 예측 결과 시각화 ↓



Prophet 최적 모델 예측 구성 요소 시각화 ↓



연월	예측값
2024-01	6097.461701
2024-02	6014.072675
2024-03	7402.700334
2024-04	4575.207735
2024-05	4499.483598
2024-06	7489.729887

<2데이터>

- 상관분석을 진행하지 않은 데이터 셋

① 최적의 하이퍼 파라미터

- changepoint_prior_scale: 0.01
- seasonality_prior_scale: 5.0
- seasonality_mode: 'multiplicative'

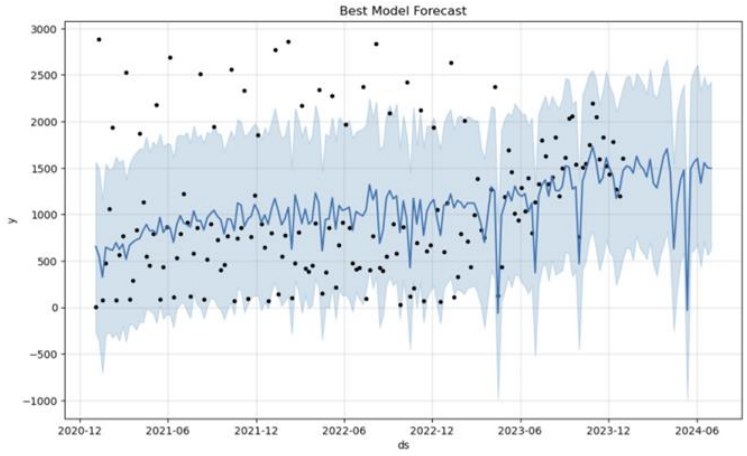
② 성능지표

MAE: 903.0364377730265

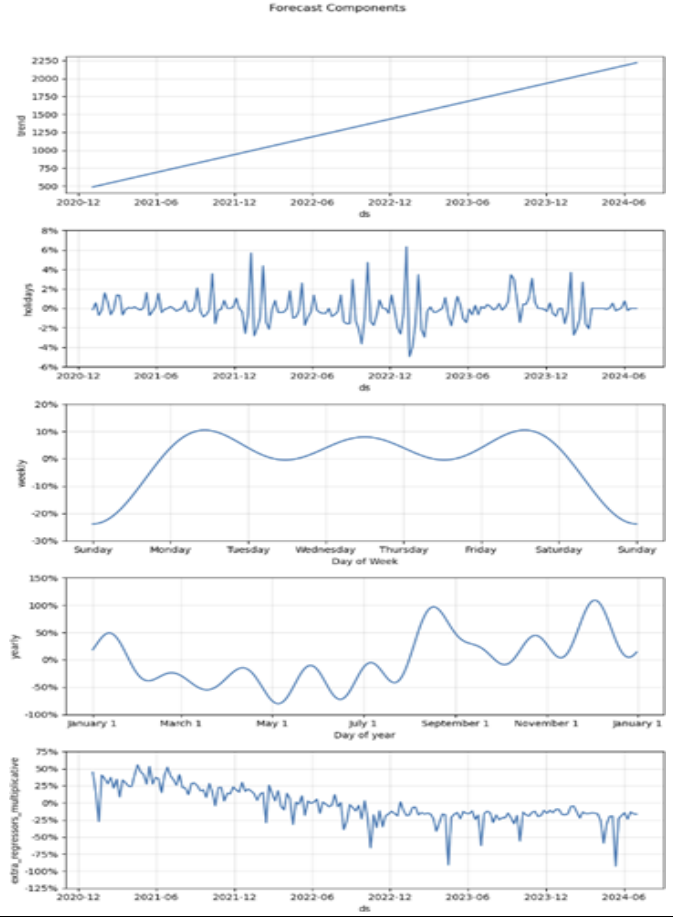
RMSE: 1015.4644964820613

③ 예측결과

Prophet 최적 모델 예측 결과 시각화 ↓



Prophet 최적 모델 예측 구성 요소 시각화 ↓



연월	예측값
2024-01	6929.013081
2024-02	6885.625769
2024-03	8488.867371
2024-04	6109.006580
2024-05	6216.076219
2024-06	8287.848968

2. SARIMA

D. 최적의 하이퍼파라미터 및 예측 결과

<1데이터>

- 상관분석을 진행한 데이터셋과 진행하지 않은 결과가 동일함

① 최적의 하이퍼파라미터

- ARIMA(5, 1, 1)
  - p (자기회귀차수): 5
  - d (차분차수): 1
  - q (이동평균차수): 1
- SARIMA(0, 0, 0)(0, 0, 0, 12)
  - P (계절적 자기회귀차수): 0
  - D (계절적 차분차수): 0
  - Q (계절적 이동평균차수): 0

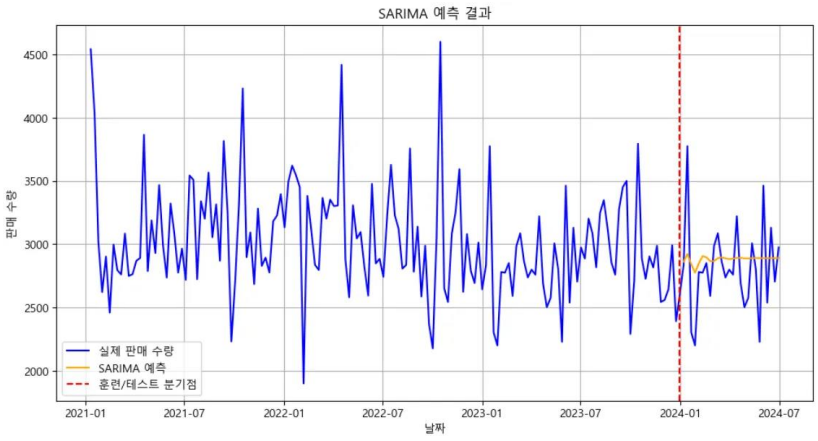
② 성능지표

MAE: 259.7137899976186

RMSE: 337.0659318165479

③ 예측결과

SARIMA 최적 모델 예측 결과 시각화 ↓



연월	예측값
2024-01	11400.088172
2024-02	11516.224940
2024-03	14421.701715
2024-04	11557.576488
2024-05	11558.170552
2024-06	14448.535270

<2데이터>

- 상관분석을 진행한 데이터셋과 진행하지 않은 결과가 동일함

① 최적의 하이퍼파라미터

- ARIMA(5, 1, 1)
  - p (자기회귀차수): 5
  - d (차분차수): 1
  - q (이동평균차수): 3
- SARIMA(0, 0, 0)(0, 0, 0, 12)
  - P (계절적 자기회귀차수): 0
  - D (계절적 차분차수): 0
  - Q (계절적 이동평균차수): 0

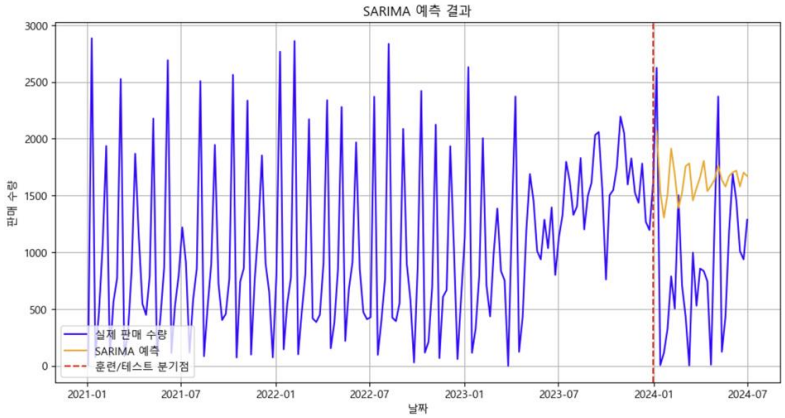
② 성능지표

MAE: 867.57265333297

RMSE: 984.1226623551711

③ 예측결과

SARIMA 최적 모델 예측 결과 시각화 ↓



연월	예측값
2024-01	6414.448374
2024-02	6510.342429
2024-03	8223.043105
2024-04	6569.267412
2024-05	6650.839129
2024-06	8374.430866

3. LSTM

C. 최적의 하이퍼파라미터 및 예측 결과

I. MAE(Mean Absolute Error)와 RMSE(Root Mean Squared Error)를 사용하여 모델 성능을 측정하여, 최적의 하이퍼파라미터 조합과 해당 모델의 성능 지표 선정함.

II. LSTM 모델의 성능은 아래와 같음.

(1데이터)

상관분석 진행O - RMSE: 815.894, MAE: 761.95

상관분석 진행X - RMSE: 826.429, MAE: 740.615

(2데이터)

상관분석 진행O - RMSE: 742.573, MAE: 638.278

상관분석 진행X - RMSE: 746.754, MAE: 642.971

III. 예측하려는 기간의 실제값과 예측값을 출력한 후, 주 단위로 되어있는 판매일 변수를 월 단위로 변환 후 예측값을 합산함.

<table>
<tr><th>상관분석</th><th>1데이터</th><th>2데이터</th></tr>
<tr><td>진행</td><td>
<table>
<tr><th>판매일</th><th>실제값</th><th>LSTM 예측값</th></tr>
<tr><td>2024-01-31</td><td>13893</td><td>16691.458984</td></tr>
<tr><td>2024-02-29</td><td>11203</td><td>13785.849609</td></tr>
<tr><td>2024-03-31</td><td>11487</td><td>14219.557617</td></tr>
<tr><td>2024-04-30</td><td>13748</td><td>17639.953125</td></tr>
<tr><td>2024-05-31</td><td>11501</td><td>13721.826172</td></tr>
<tr><td>2024-06-30</td><td>11346</td><td>14148.600586</td></tr>
</table>
</td><td>
<table>
<tr><th>판매일</th><th>실제값</th><th>LSTM 예측값</th></tr>
<tr><td>2024-01-31</td><td>3859</td><td>6296.511719</td></tr>
<tr><td>2024-02-29</td><td>3146</td><td>5311.955566</td></tr>
<tr><td>2024-03-31</td><td>2385</td><td>5013.180664</td></tr>
<tr><td>2024-04-30</td><td>5231</td><td>6136.764160</td></tr>
<tr><td>2024-05-31</td><td>3432</td><td>4834.106445</td></tr>
<tr><td>2024-06-30</td><td>4687</td><td>5333.137207</td></tr>
</table>
</td></tr>
<tr><td>미진행</td><td>
<table>
<tr><th>판매일</th><th>실제값</th><th>LSTM 예측값</th></tr>
<tr><td>2024-01-31</td><td>13893</td><td>18024.990234</td></tr>
<tr><td>2024-02-29</td><td>11203</td><td>14667.341797</td></tr>
<tr><td>2024-03-31</td><td>11487</td><td>14660.949219</td></tr>
<tr><td>2024-04-30</td><td>13748</td><td>17762.513672</td></tr>
<tr><td>2024-05-31</td><td>11501</td><td>14184.575195</td></tr>
<tr><td>2024-06-30</td><td>11346</td><td>14698.484375</td></tr>
</table>
</td><td>
<table>
<tr><th>판매일</th><th>실제값</th><th>LSTM 예측값</th></tr>
<tr><td>2024-01-31</td><td>3859</td><td>6690.575684</td></tr>
<tr><td>2024-02-29</td><td>3146</td><td>5702.192383</td></tr>
<tr><td>2024-03-31</td><td>2385</td><td>5469.315430</td></tr>
<tr><td>2024-04-30</td><td>5231</td><td>6633.763184</td></tr>
<tr><td>2024-05-31</td><td>3432</td><td>5203.310547</td></tr>
<tr><td>2024-06-30</td><td>4687</td><td>5669.668945</td></tr>
</table>
</td></tr>
</table>

4. GRU

C. 최적의 하이퍼파라미터 및 예측 결과

I. MAE(Mean Absolute Error)와 RMSE(Root Mean Squared Error)를 사용하여 모델 성능을 측정하여, 최적의 하이퍼파라미터 조합과 해당 모델의 성능 지표 선정함.

II. GRU 모델의 성능은 아래와 같음.

(1데이터)

상관분석 진행O - RMSE: 704.427, MAE: 656.506

상관분석 진행X - RMSE: 772.813, MAE: 700.922

(2데이터)

상관분석 진행O - RMSE: 714.904, MAE: 608.05

상관분석 진행X - RMSE: 702.377, MAE: 595.325

III. 예측하려는 기간의 실제값과 예측값을 출력한 후, 주 단위로 되어있는 판매일 변수를 월 단위로 변환 후 예측값을 합산함.

<table>
<tr><th>상관분석</th><th>1데이터</th><th>2데이터</th></tr>
<tr><td>진행</td><td>
<table>
<tr><th>판매일</th><th>실제값</th><th>GRU 예측값</th></tr>
<tr><td>2024-01-31</td><td>13893</td><td>16671.042969</td></tr>
<tr><td>2024-02-29</td><td>11203</td><td>13790.599609</td></tr>
<tr><td>2024-03-31</td><td>11487</td><td>14053.469727</td></tr>
<tr><td>2024-04-30</td><td>13748</td><td>17431.302734</td></tr>
<tr><td>2024-05-31</td><td>11501</td><td>13625.105469</td></tr>
<tr><td>2024-06-30</td><td>11346</td><td>13903.893555</td></tr>
</table>
</td><td>
<table>
<tr><th>판매일</th><th>실제값</th><th>GRU 예측값</th></tr>
<tr><td>2024-01-31</td><td>3859</td><td>5700.888672</td></tr>
<tr><td>2024-02-29</td><td>3146</td><td>4577.342285</td></tr>
<tr><td>2024-03-31</td><td>2385</td><td>4413.018555</td></tr>
<tr><td>2024-04-30</td><td>5231</td><td>5469.509766</td></tr>
<tr><td>2024-05-31</td><td>3432</td><td>4345.666992</td></tr>
<tr><td>2024-06-30</td><td>4687</td><td>4731.109375</td></tr>
</table>
</td></tr>
<tr><td>미진행</td><td>
<table>
<tr><th>판매일</th><th>실제값</th><th>GRU 예측값</th></tr>
<tr><td>2024-01-31</td><td>13893</td><td>17665.380859</td></tr>
<tr><td>2024-02-29</td><td>11203</td><td>14423.778320</td></tr>
<tr><td>2024-03-31</td><td>11487</td><td>14430.361328</td></tr>
<tr><td>2024-04-30</td><td>13748</td><td>17547.830078</td></tr>
<tr><td>2024-05-31</td><td>11501</td><td>13987.624023</td></tr>
<tr><td>2024-06-30</td><td>11346</td><td>14371.171875</td></tr>
</table>
</td><td>
<table>
<tr><th>판매일</th><th>실제값</th><th>GRU 예측값</th></tr>
<tr><td>2024-01-31</td><td>3859</td><td>6087.492676</td></tr>
<tr><td>2024-02-29</td><td>3146</td><td>4911.905762</td></tr>
<tr><td>2024-03-31</td><td>2385</td><td>4798.817383</td></tr>
<tr><td>2024-04-30</td><td>5231</td><td>5930.996582</td></tr>
<tr><td>2024-05-31</td><td>3432</td><td>4704.814453</td></tr>
<tr><td>2024-06-30</td><td>4687</td><td>4992.263672</td></tr>
</table>
</td></tr>
</table>

[모델별 성능 비교 및 최종 모델 선정]

1데이터와 2데이터에 대하여 상관분석을 진행한 경우, 진행하지 않은 경우 두 가지 케이스에서 더 좋은 성능을 보인 지표를 선정하였고, 모델별 성능을 비교함. 그 결과 1데이터는 SARIMA, 2데이터는 상관분석을 진행하지 않은 GRU가 최종 모델로 선정됨.

<1데이터>

	RMSE	MAE
Prophet	345.62705989119394	267.3775352975817
SARIMA	337.0659318165479	259.7137899976186
LSTM	815.8942885682557	761.9501207139757
GRU	704.4273322015736	656.5061102973091

<2데이터>

	RMSE	MAE
Prophet	837.9476429563201	710.9824727034639
SARIMA	984.1226623551711	867.57265333297
LSTM	742.573467226079	638.2780512029475
GRU	702.3774340558903	595.32504827326

각 모델의 2024년도 1월부터 6월 1데이터의 '(중분류) 라면, 통조림, 상온즉석' 판매수량 예측값과 2데이터의 '(대분류) 면류, 라면류' 판매수량 예측값은 아래와 같음.

판매일	1데이터(SARIMA)	2데이터(GRU)
2024.01	11400.088172	6087.492676
2024.02	11516.224940	4911.905762
2024.03	14421.701715	4798.817383
2024.04	11557.576488	5930.996582
2024.05	11558.170552	4704.814453
2024.06	14448.535270	4992.263672

◦ 활용방안

본 보고서에서 제시한 예측 모델은 다양한 활용 방안을 통해 도매 물류센터의 운영 효율성을 극대화할 수 있음.

첫째, 재고관리 최적화가 가능함. 이 모델은 제품 간의 상호 연관성을 분석해 재고 관리 전략을 개선하는 데 기여할 수 있음. 목적에 맞게 데이터의 중분류 혹은 대분류 타겟을 다르게 적용 가능함. 이를 통해 관련 제품의 동반 발주 가능성을 예측할 수 있고 재고 최적화를 도모함.

둘째, 효율적인 공급 계획 수립이 가능함. 주차별 연관분석을 바탕으로 발주되는 제품 조합을 예측함으로써 보다 정교한 공급 계획을 수립할 수 있음. 이는, 발주 패턴을 반영하여 적합한 시점에 제품을 공급하고, 공급망 내의 불필요한 지연을 줄임으로써 운영 효율성을 높일 수 있음.

셋째, 시즌별 수요 예측이 가능함. 주차별 발주 데이터를 축적함으로써 장기적으로 특정 시즌에 수요가 급증하거나 감소하는 패턴을 예측할 수 있음. 이를 통해, 도매 물류센터는 계절적 수요 변화에 빠르게 대응할 수 있으며, 미래 수요 예측에 기반한 리스크 관리를 통해 효율적인 공급망 운영이 가능함.

넷째, 경제 지표를 추가하여 정교한 분석이 가능함. 소비자 경제에 영향을 미치는 다양한 경제 지수를 변수로 활용하였음. 추후 고려하지 못했던 가계 경제 지수 등의 세부적인 경제 지표 및 원화 환율 등을 추가 변수로 고려한다면, 더욱 정확한 예측이 가능할 것으로 보임.

이번 연구로 3개년의 제한적인 데이터로 미래 6개월을 예측하는 단편적인 결과를 낳았음. 하지만 이후 장기간에 걸쳐 빅데이터가 누적되면 될수록 기간적인 예측에 있어 더 정확한 예측이 가능하며 이 결과로 리스크를 최소한으로 줄일 수 있을 것으로 기대됨. 향후 후속 연구에 있어서는 경제 지수 외에 더 다양한 요인들을 추가한다면 더 의미 있는 연구가 될 것이라 기대됨.

◦ 활용데이터 및 참고 문헌 출처 등

데이터명	출처	데이터명	출처
중소유통물류센터 거래 1데이터	주최 기관	중소유통물류센터 거래 2데이터	주최기관
소비자물가지수	통계청	소비자 심리지수	e-나라지표
생활물가지수	통계청	국내총생산(명목 GDP)	e-나라지표
전체 실업률	통계청	경제성장률(실질 GDP)	e-나라지표
남성 실업률	통계청	환율(원/달러)	e-나라지표
여성 실업률	통계청	기준금리	e-나라지표

허침(2023). **연관 분석 알고리즘의 분석 및 개선 : Apriori 알고리즘 개선을 중심으로**. 서울 : 건국대학교 대학원 석사학위논문.

강민수, 박상현, 김태균, 이태희(2023). **SARIMA 모델을 이용한 비선형 시계열 자료의 단기 예측**. 대한기계학회.

제3회 유통데이터 활용 경진대회
(수요예측 부문) 분석보고서 요약서

1. 팀명 : FC BOAZ

2. 세부내용

ㅇ 분석 주제

Apriori 연관 분석을 활용한 특정 제품군 수요 예측 모델

ㅇ 분석 목적

본 연구는 연관분석을 통해 도매점에서 발주되는 제품 간의 상관관계를 분석하여 1데이터의 경우 '라면, 통조림, 상온즉석' 중분류 제품군을, 2데이터의 경우 '면류, 라면류' 대분류 제품군의 수요를 예측하는 것을 목표로 함. 이를 통해 제품 간 상호 연관성을 바탕으로 정확한 수요 예측을 수행하고, 재고 관리 및 공급 계획을 최적화할 수 있음. 궁극적으로, 도매점이 발주 패턴에 따른 수요 변동에 신속하게 대응하여 유통망의 운영 효율성을 추구하는 것이 목적임.

ㅇ 핵심 내용

- 1 데이터와 2 데이터의 분포를 파악하기 위해 EDA를 진행
- '반품'데이터 처리를 위해 판매일과 매출처코드별로 판매수량을 집계
- 옵션코드별로 판매수량 집계가 상이하기 때문에 옵션코드(BX, CS, EA) 값 기준으로 데이터를 분할하여 판매수량 이상치 처리(IQR 사분위, Standard Deviation Method, Z-score Method, Isolation Forest 네 가지 방식 사용)
- Apriori 연관 분석을 통한 타겟 제품군과 타 제품과의 관계 파악 및 신뢰도 컬럼 추가
- 연월별 경제 지표 / 공공데이터 변수를 추가(소비자물가지수, 생활물가지수, 전체 실업률, 남성 실업률, 여성 실업률, 소비자 심리 지수, 국내총생산, 경제성장률, 환율, 기준금리)
- 판매수량과의 상관계수 절댓값이 0.1 이상인 독립변수만 남기고, 나머지 변수는 삭제. 이 과정을 통해 두 개의 데이터셋(삭제된 데이터셋과 삭제되지 않은 데이터셋)을 생성
- Prophet, SARIMA 시계열 모델과 LSTM, GRU 딥러닝 모델을 활용해 예측 진행
- 각 모델의 MAE, RMSE 성능 지표를 비교하여 최적 모델 선정 및 최종 예측값 산출

ㅇ 기대효과

첫째, 재고 관리 최적화를 통해 관련 제품의 동반 발주 가능성을 예측하고, 재고 부족이나 과잉을 방지할 수 있음. 둘째, 시즌별 수요 예측을 통해 장기적으로 계절적 수요 변화에 빠르게 대응하고 리스크를 최소화하며 운영 효율성을 높일 수 있음. 셋째, 경제 지표 추가 분석을 통해 경제적 변수들을 활용하여 더욱 정교하고 정확한 예측이 가능해질 것임.

ㅇ 활용데이터 및 참고 문헌 출처

주최 기관 : 중소유통물류센터 거래 데이터 - 유통데이터 활용 경진대회 배포용 1 & 2 데이터

통계청 : 소비자물가지수, 생활물가지수, 실업률(전체,남성,여성)

e-나라지표 : 소비자심리지수, 국내총생산(명목GDP), 경제성장률(실질GDP), 환율, 기준금리

허침(2023). **연관 분석 알고리즘의 분석 및 개선 : Apriori 알고리즘 개선을 중심으로**. 서울 : 건국대학교 대학원 석사학위논문.

강민수, 박상현, 김태균, 이태희(2023). **SARIMA 모델을 이용한 비선형 시계열 자료의 단기 예측**. 대한기계학회.