Capstone 1.
매출 영향 특성 항목 찾기

3조
이지호, 정서영, 이건영, 윤지현

0. 사용한 Tool과 Python Library

- 사용한 Tool

시각화, 데이터 분석 : Tableau, Notebook(Jupyter, Colab 등)

- 사용한 Python Library

📊 Data / Preprocessing / Visualization
🐼 pandas — 표 데이터 처리
🔢 numpy — 수치·배열 연산
🧪 scipy — 과학 계산(통계/최적화)
🔤 re — 정규식 기반 문자열 처리
📉 matplotlib.pyplot — 기본 그래프 시각화
🎨 seaborn — 통계적 시각화

🤖 Machine Learning (Scikit-learn)
✂️ train_test_split — 학습/테스트 데이터 분리
⚖️ StandardScaler / MinMaxScaler — 스케일링
🔠 LabelEncoder — 범주형 인코딩
🧬 MLPClassifier — 신경망 분류기
📏 MSE / MAE / R^2 — 회귀 평가 지표
📊 permutation_importance — 변수 중요도
🌲 RandomForestRegressor — 트리 기반 앙상블
🌳 ExtraTreesRegressor — 랜덤성 강화 트리 모델

🚀 Boosting Models
🔥 XGBoost (XGBRegressor) — 고성능 GBDT 회귀
⚡ LightGBM — 빠르고 효율적인 부스팅
  • early_stopping — 과적합 방지
  • log_evaluation — 학습 로그 출력
🐱 CatBoost — 범주형 자동처리 부스팅
  • CatBoostRegressor
  • Pool

🧠 Deep Learning
🟠 PyTorch
torch — 텐서 및 모델 구성
torch.nn — 신경망 모듈
torch.optim — 옵티마이저
📉 mse_loss — 회귀 손실 함수
🔵 TensorFlow / Keras
Sequential — 모델 구조
LSTM — 시계열 모델
Dense — fully connected layer
EarlyStopping — 조기 종료

🔍 Explainability
💡 SHAP — 모델 예측 원인 분석
🎯 Captum — PyTorch 모델 해석

🖥️ Environment
💻 platform — OS/환경 정보
⚙️ sys — 시스템 경로/환경
🗂️ google.colab.drive — 드라이브 연동

1. 데이터 탐색

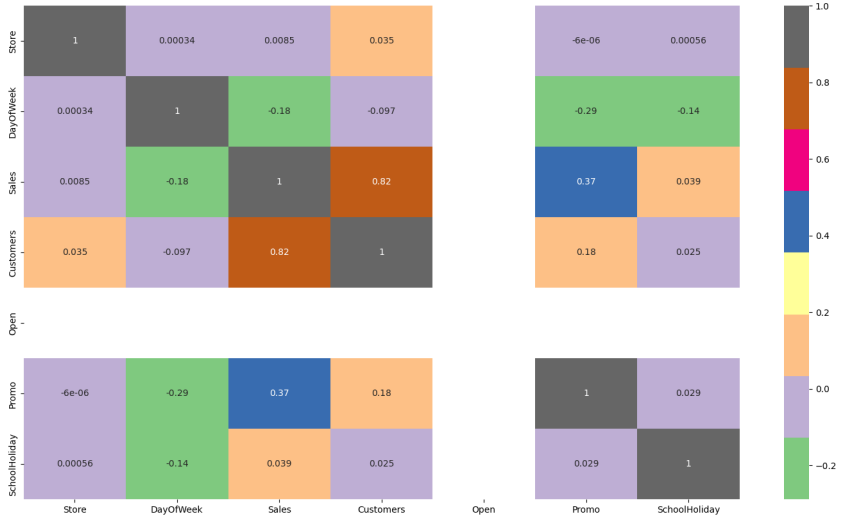
store.csv와 train.csv 데이터의 정보

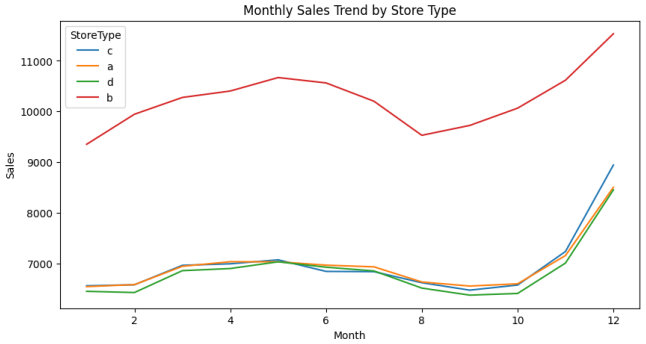
컬럼명	설명
Store	매장 ID(고유 식별자)
DayOfWeek	요일(1=월요일 ~ 7=일요일)
Date	판매 발생 날짜 (YYYY-MM-DD)
Sales	해당 날짜의 매출(예측 대상)
Customers	방문 고객 수
Open	매장 오픈 여부 (0=휴무, 1=영업)
Promo	프로모션 진행 여부 (1=진행중)
StateHoliday	공휴일 여부 (0=평일, a=국경일, b=부활절, c=성탄절)
SchoolHoliday	학교 휴일 여부 (1=방학 등)

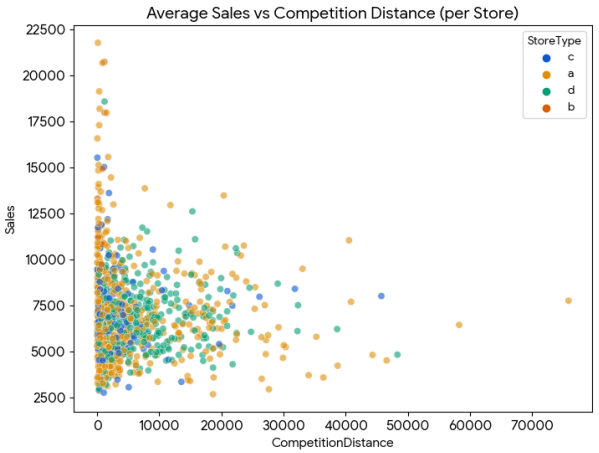
컬럼명	설명
Store	매장 ID (train과 연결되는 key)
StoreType	매장 유형 (a=드럭스토어,b=대형마트,c=할인형 디스카운트 매장, d=특수매장)
Assortment	상품 구성 폭 (a=기본, b=확장, c=대형 구성)
CompetitionDistance	경쟁사 매장까지 거리(m)
CompetitionOpenSinceMonth	경쟁사 매장이 오픈한 월
CompetitionOpenSinceYear	경쟁사 매장이 오픈한 연도
Promo2	지속 프로모션 참여 여부 (0/1)
Promo2SinceWeek	Promo2가 시작된 주차
Promo2SinceYear	Promo2가 시작된 연도
PromoInterval	매년 반복 프로모션이 활성화되는 월 (예: "Jan,Apr,Jul,Oct")

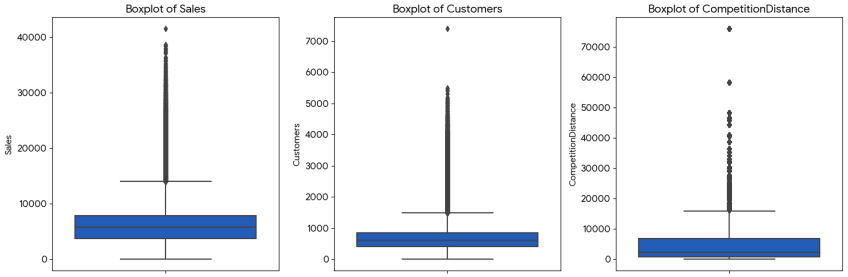
1. 데이터 탐색

store.csv와 train.csv 데이터를 각자 자유롭게 분석하면서 insignt와 전처리 방법을 논의함









- 1사람당 9.5개씩 구매를 하는 경향이 보임 → 대형 마트에서의 대량 구매 패턴같음
- boxplot상에서 sales 값이 15,000 이상일 경우 이상치라고 답하며 전체 매출 분포의 1/20 수준임(80만건 중에 2.5만건) -> 실제 영업 이벤트로 발생하는 정상적 매출이며, 이때의 조건이 상위 매출을 좌우한다고 볼 수 있음
- 공휴일은 고객수가 늘어나면서 매출도 높음
- B 타입 매장이 매출이 높음 (…)

1. 데이터 탐색

각자의 store.csv와 train.csv 분석 결과를 바탕으로 결측치 및 이상치 처리 방안을 논의함

(1) 결측치 확인 및 처리

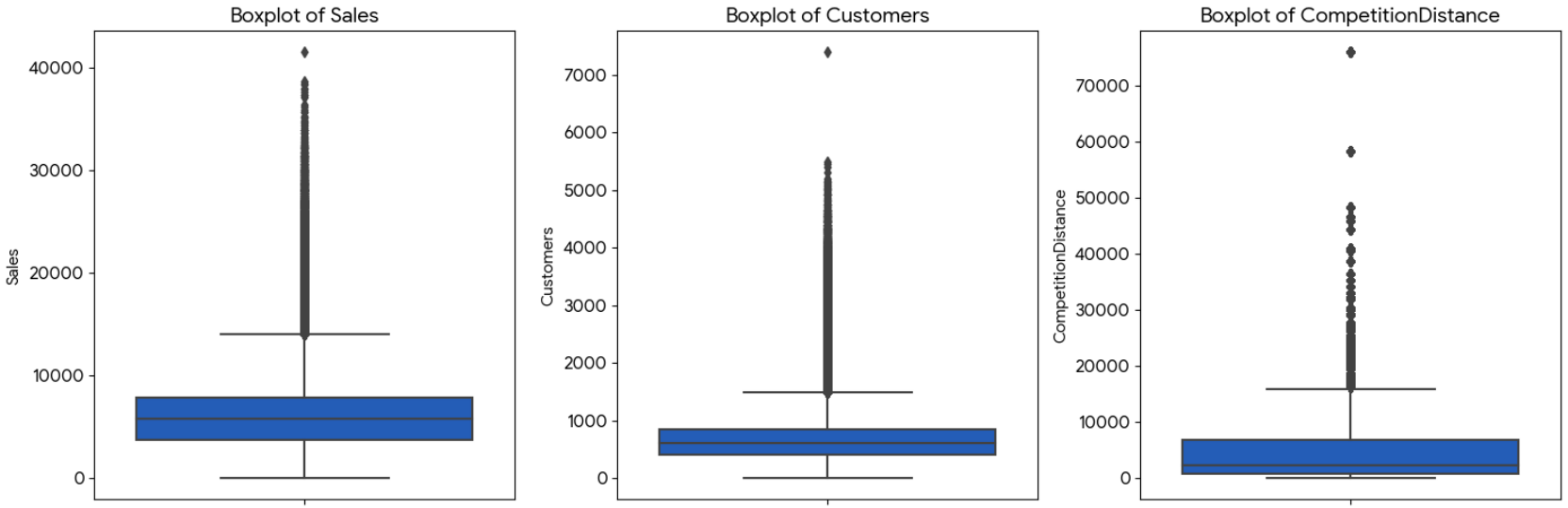
- open 칼럼의 값이 '0'일 경우 매출도 0이므로 의미가 없을 것이라 여겨 삭제처리 하기로 함
- 291, 622, 879 매장은 인근 경쟁 매장이 없어 분석 대상에서 제외 (Competition Distance가 null)
- Promo2가 0일 경우 promo2 관련 데이터가 모두 null → null이라고 해서 전부 삭제할 필요는 없고 '프로모션이 없음'이라는 하나의 '정보로 간주'
- Sales(매출액)이 0인 경우 삭제 처리

2. 데이터 전처리

각자의 store.csv와 train.csv 분석 결과를 바탕으로 결측치 및 이상치 처리 방안을 논의함

(2) 이상치 확인 및 처리

- boxplot상에서 sales 값이 15,000 이상일 경우 이상치로 분석됨 → 전체 매출 분포의 1/20 수준임(80만건 중에 2.5만건) -> 실제 영업 이벤트로 발생하는 정상적 매출이며, 이때의 조건이 상위 매출을 좌우한다고 볼 수 있으므로 그대로 두기로 함



2. 데이터 전처리

store.csv와 train.csv 데이터셋을 merge 하고 숫자형 파생 컬럼을 생성

(3) Data merge 및 숫자형 데이터 컬럼 생성 - 1

- store.csv를 기준으로 train.csv 데이터를 merge하고 StateHoliday, Storetype, Assortment 등을 숫자형으로 변형

```
1.# Column Non-Null Count Dtype
--- ------ -------------- -----
0 Store 1017209 non-null int64
1 DayOfWeek 1017209 non-null int64
2 Date 1017209 non-null object 3 Sales 1017209 non-null int64
4 Customers 1017209 non-null int64
5 Open 1017209 non-null int64
6 Promo 1017209 non-null int64
7 StateHoliday 1017209 non-null object
8 SchoolHoliday 1017209 non-null int64
9 StoreType 1017209 non-null object
10 Assortment 1017209 non-null object
11 CompetitionDistance 1014567 non-null float64
12 CompetitionOpenSinceMonth 693861 non-null float64
13 CompetitionOpenSinceYear 693861 non-null float64
14 Promo2 1017209 non-null int64
15 Promo2SinceWeek 509178 non-null float64
16 Promo2SinceYear 509178 non-null float64
17 PromoInterval 509178 non-null object
dtypes: float64(5), int64(8), object(5) memory usage: 139.7+ MB
```

2. 데이터 전처리

store.csv와 train.csv 데이터셋을 merge 하고 숫자형 파생 컬럼을 생성

(3) Data merge 및 숫자형 데이터 컬럼 생성 - 2

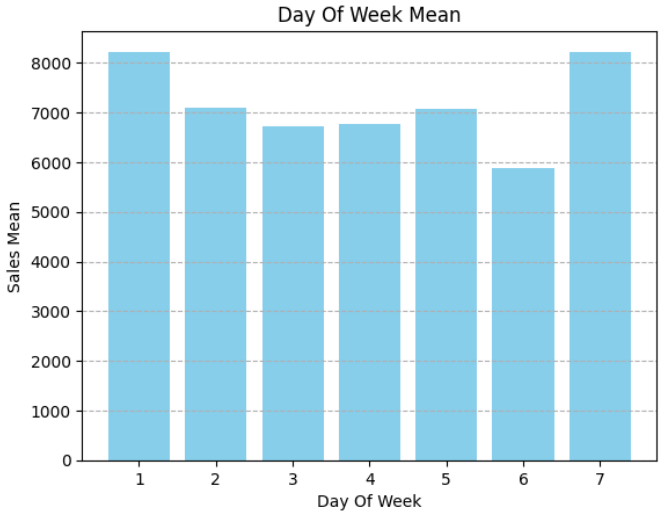
- Merge 한 데이터셋에 Data, Promo2 등을 파생 컬럼을 생성하여 데이터처리

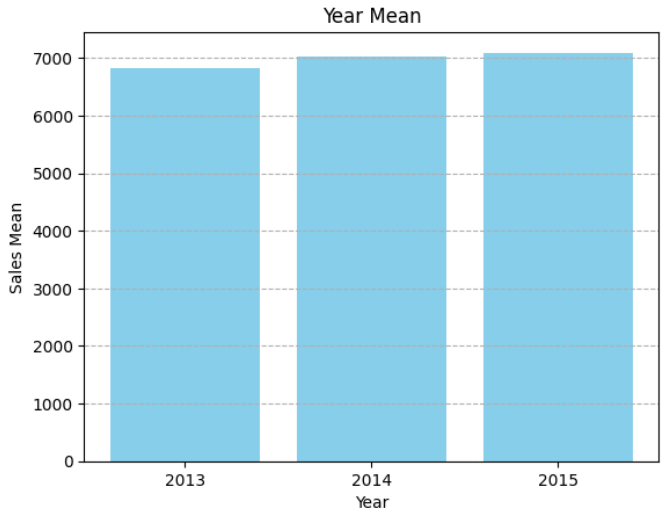
컬럼명	설명	컬럼명	설명
Store	매장 고유 번호	Year	연도
DayOfWeek	요일 (1~7)	Month	월 (1~12)
Sales	매출값	Day	일 (1~31)
Customers	고객 수	WeekOfYear	ISO 기준 주차
Open	영업 여부	Quarter	분기 (1~4)
Promo	당일 프로모션 여부	IsMonthStart	월 시작 여부
StateHoliday	공휴일 여부 (0, a, b, c → 숫자 인코딩됨)	IsMonthEnd	월 말 여부
SchoolHoliday	학교 휴일 여부	IsYearStart	1월 1일 여부
StoreType	매장 유형 (인코딩된 숫자)	IsYearEnd	12월 31일 여부
Assortment	상품 구성 범위 (인코딩된 숫자)	CompetitionOpenDuration	경쟁사 매장 오픈 후 지난 일수
CompetitionDistance	경쟁사 거리	Promo2OpenDuration	지속 프로모션(Promo2) 시작 후 지난 일수
CompetitionOpenSinceMonth	경쟁사 오픈 월		
CompetitionOpenSinceYear	경쟁사 오픈 연도		
Promo2	지속 프로모션 참여 여부		
Promo2SinceWeek	지속 프로모션 시작 주차		
Promo2SinceYear	지속 프로모션 시작 연도		

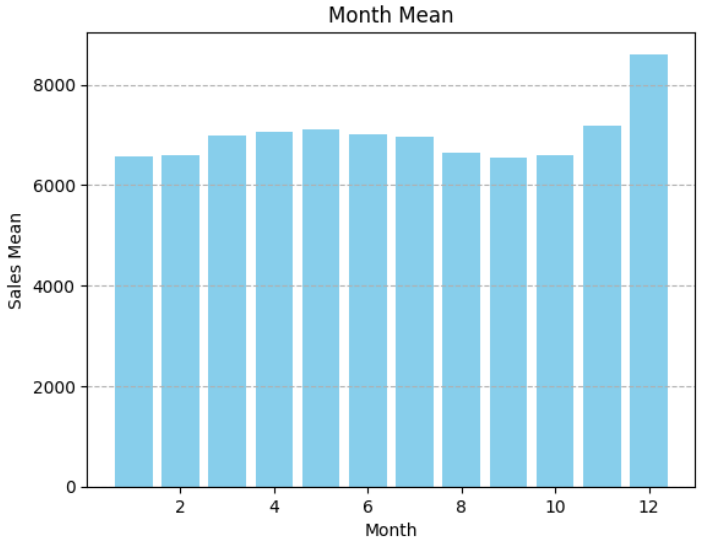
3. Insight 도출

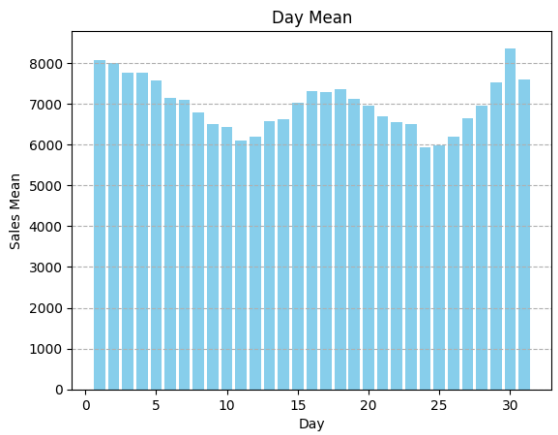
최종 Dataset의 컬럼별 Theme를 도출하여 각자 분석해보고 Theme별 Insight를 도출함

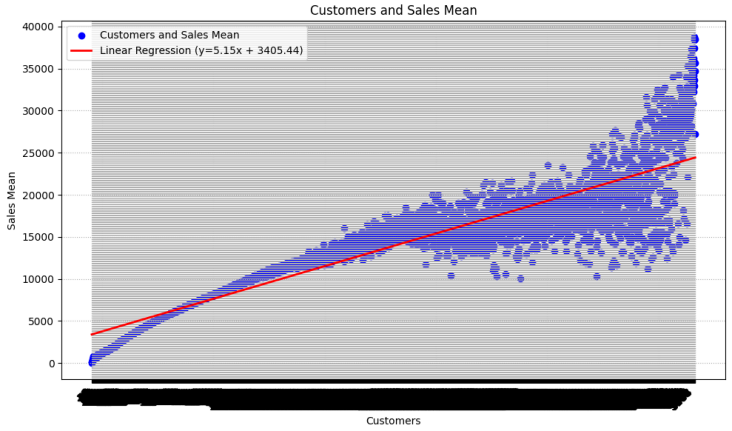
(1) Store 운영 데이터별 Insight - 1

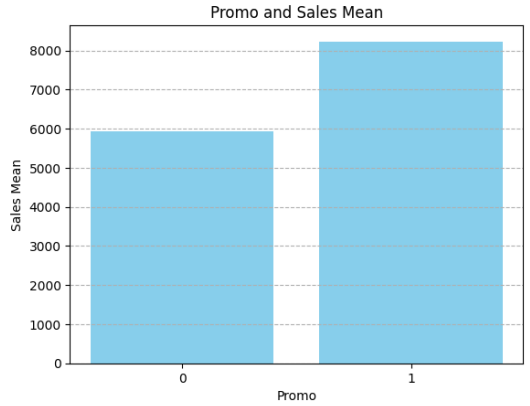








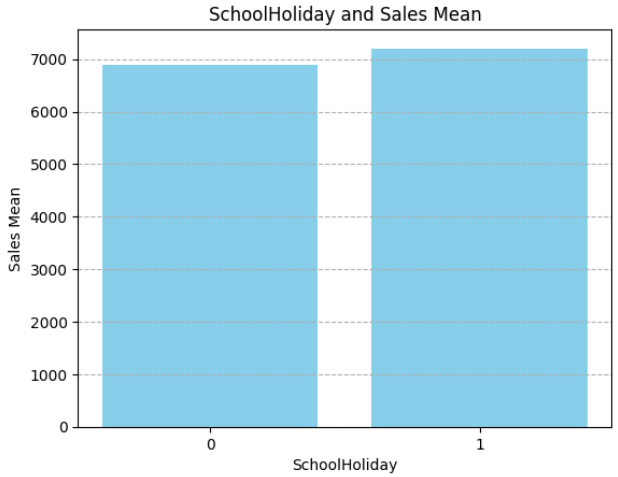


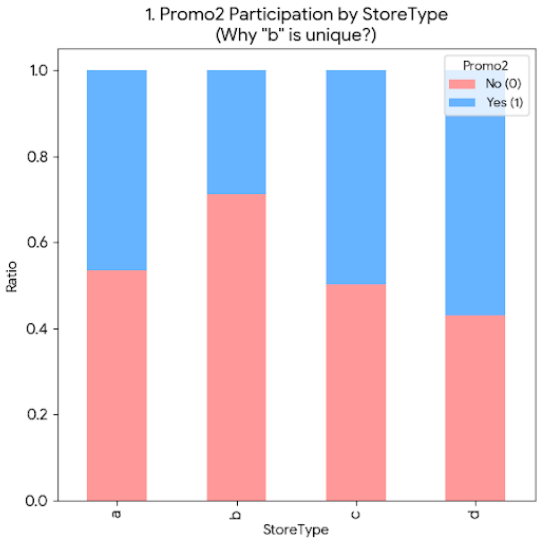


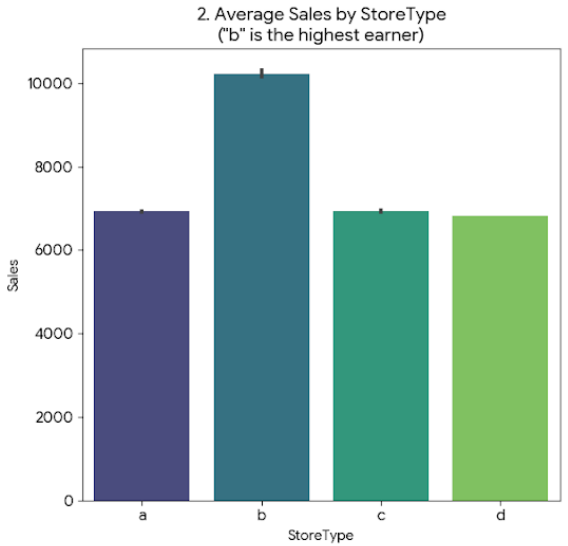
3. Insight 도출

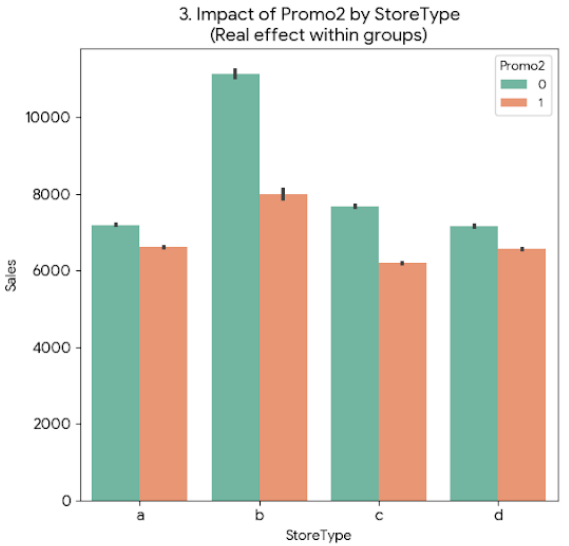
최종 Dataset의 컬럼별 Theme를 도출하여 각자 분석해보고 Theme별 Insight를 도출함

(1) Store 운영 데이터별 Insight - 2





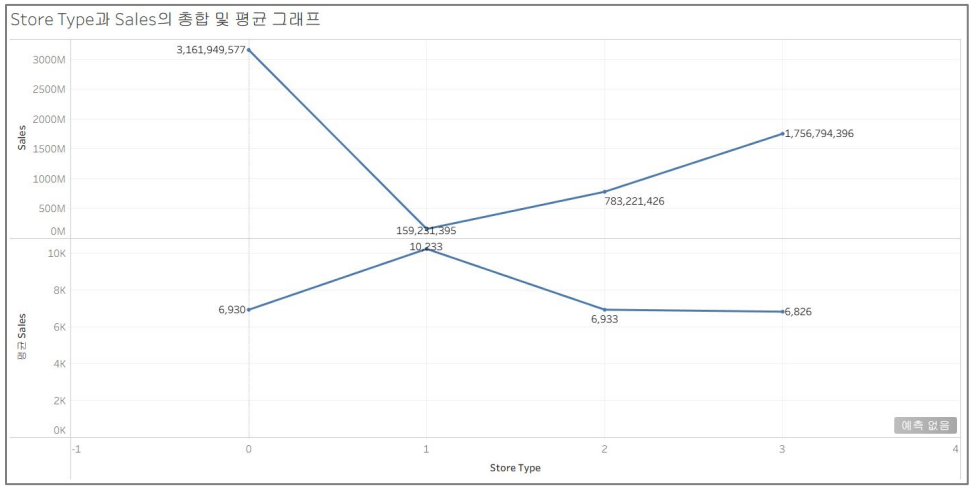


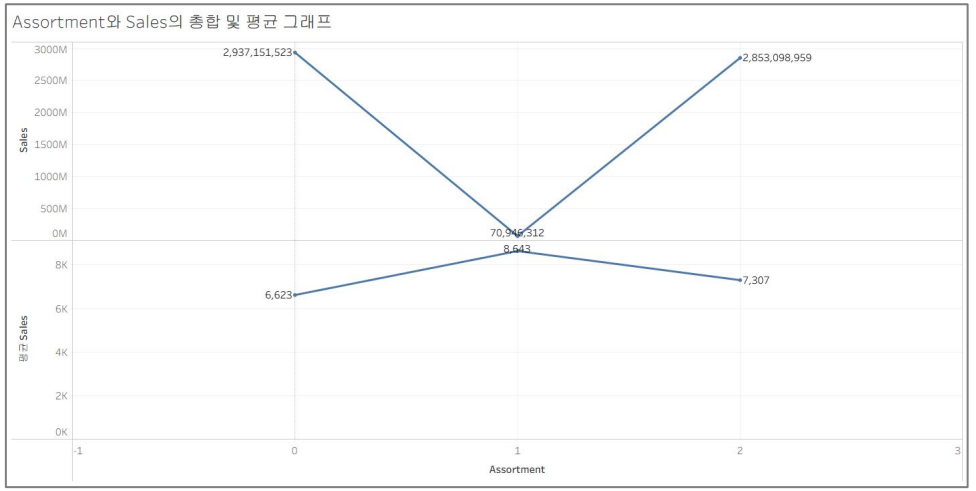


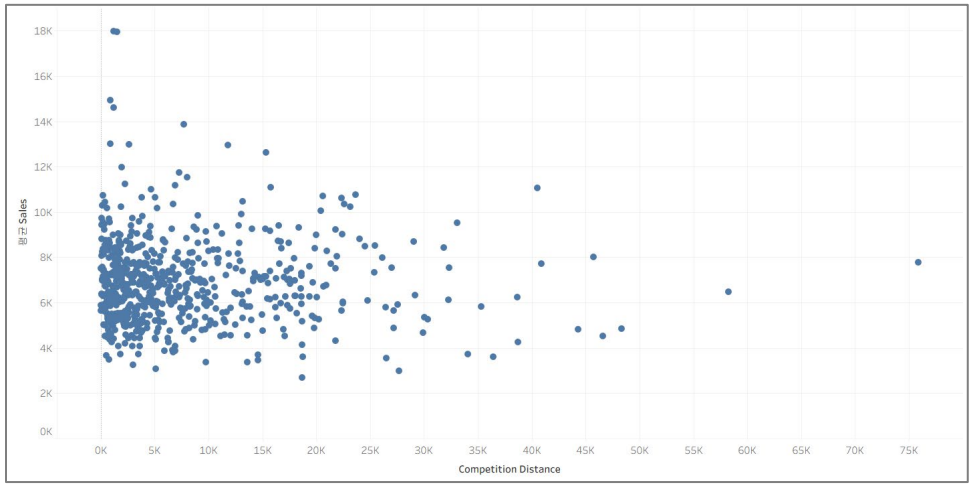
3. Insight 도출

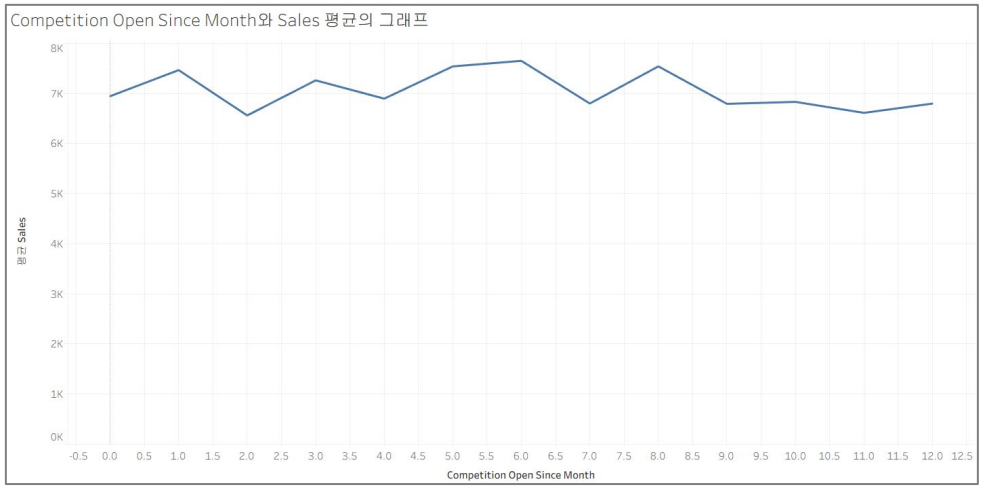
최종 Dataset의 컬럼별 Theme를 도출하여 각자 분석해보고 Theme별 Insight를 도출함

(2) 매장 고정 속성(Store.csv 기반)





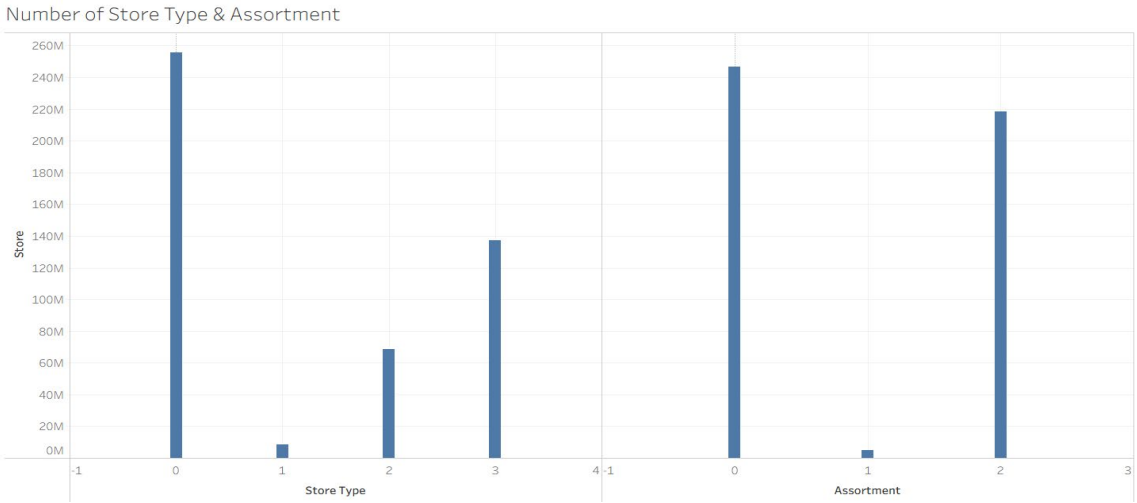


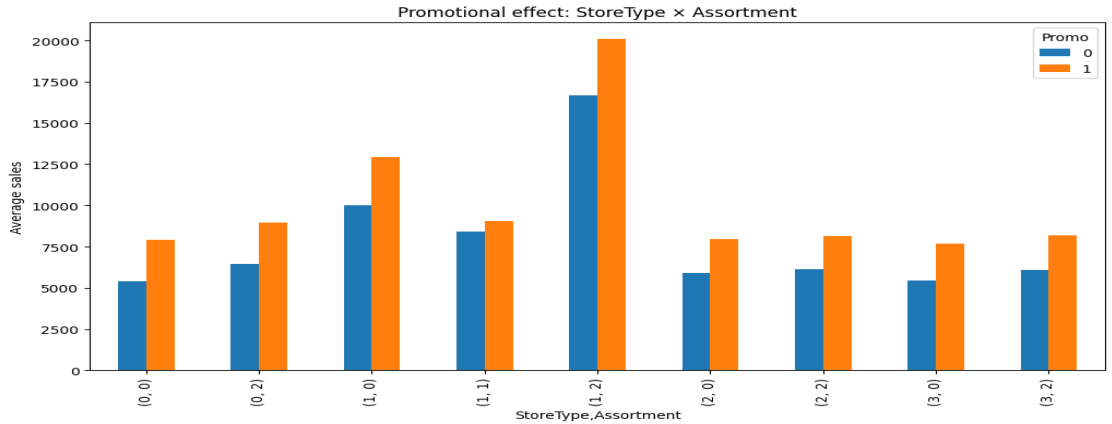


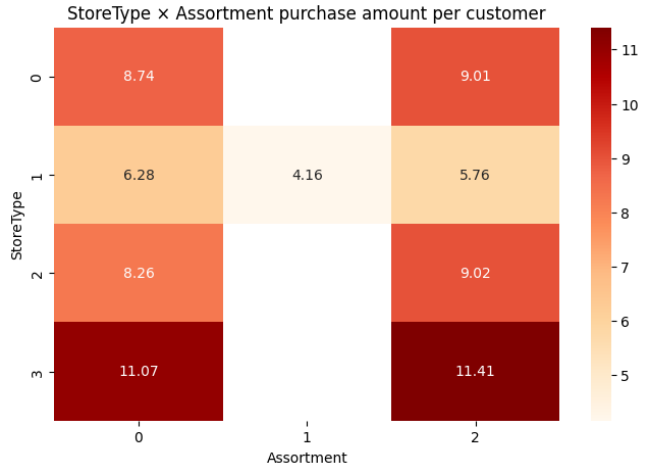
3. Insight 도출

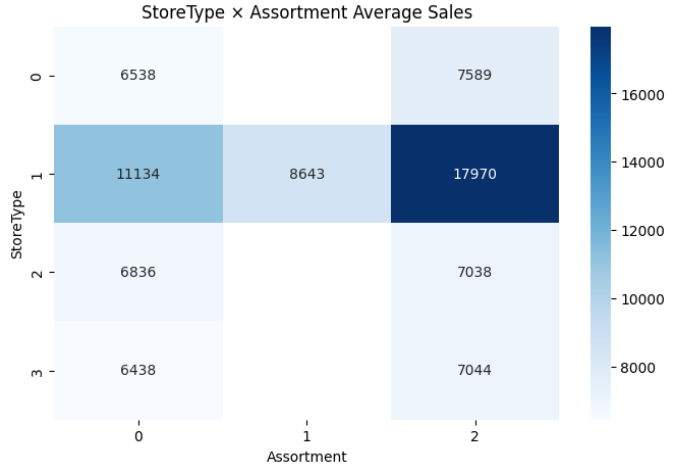
최종 Dataset의 컬럼별 Theme를 도출하여 각자 분석해보고 Theme별 Insight를 도출함

(2) 매장 고정 속성(Store.csv 기반)







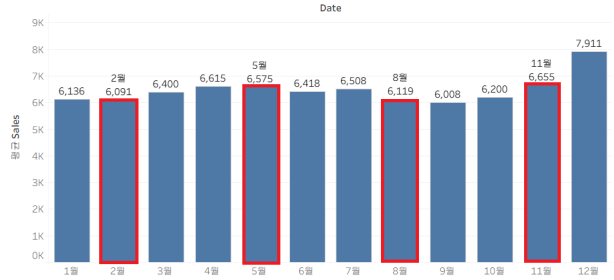


3. Insight 도출

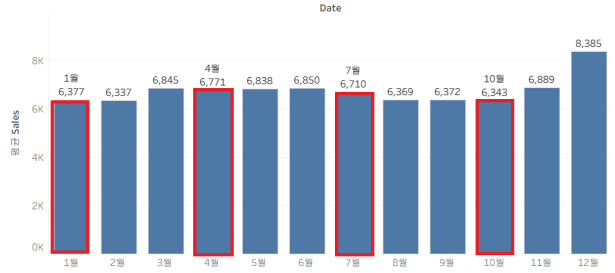
최종 Dataset의 컬럼별 Theme를 도출하여 각자 분석해보고 Theme별 Insight를 도출함

(3) 프로모션 정보

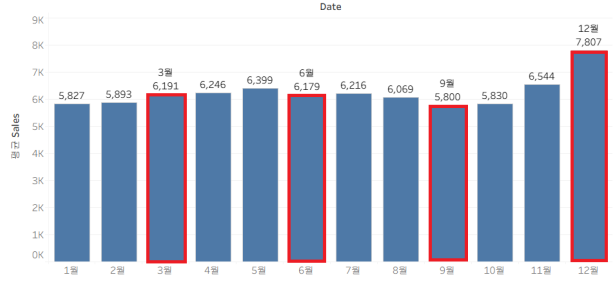
"월별 패턴" 확인하기: 2 5 8 11



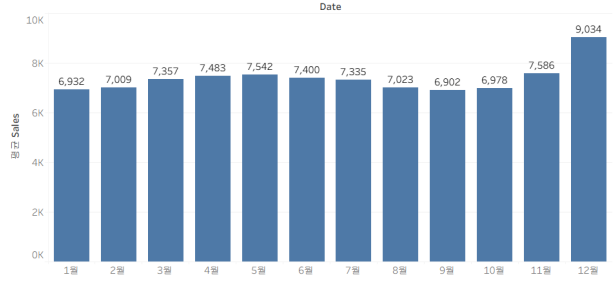
"월별 패턴" 확인하기: 1 4 7 10



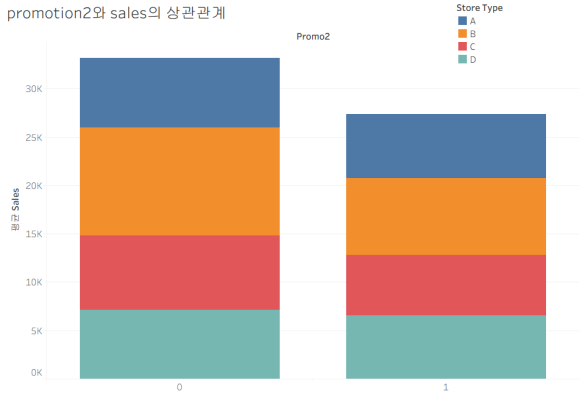
"월별 패턴" 확인하기: 3 6 9 12



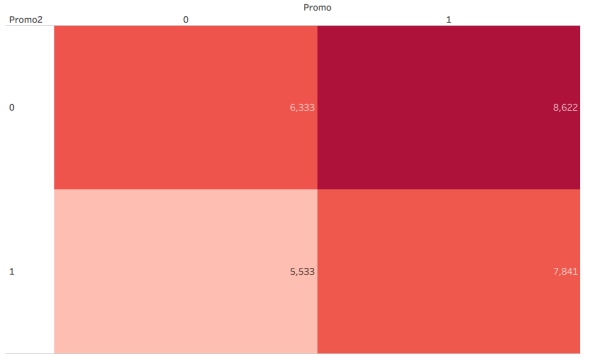
"월별 패턴" 확인하기: NaN



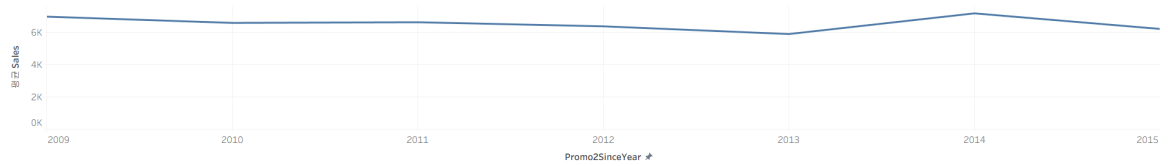
promotion2와 sales의 상관관계



프로모션별 sales 효과



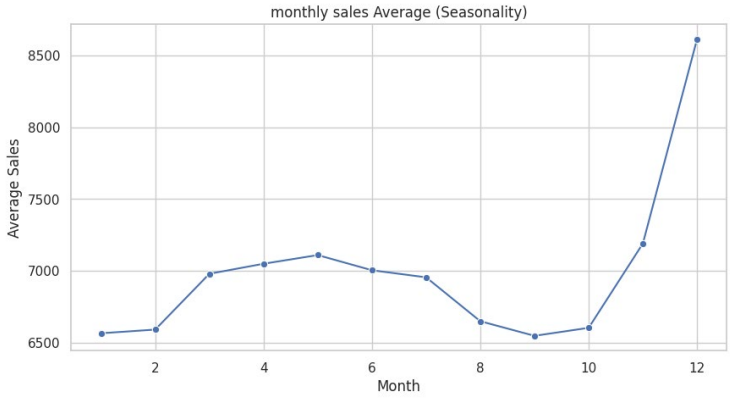
Promotion since year

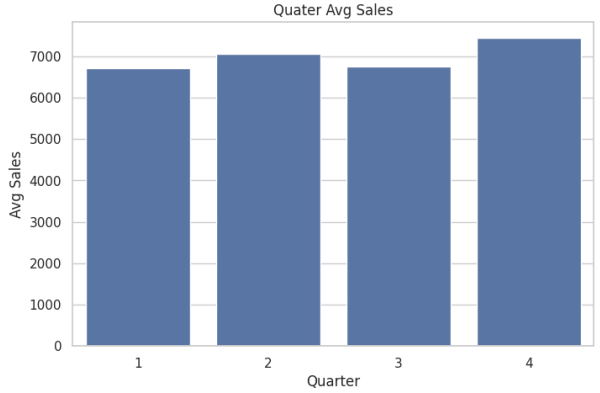


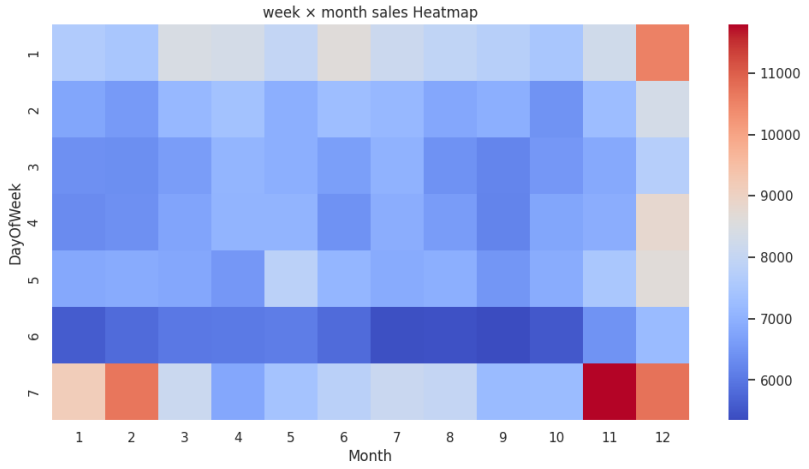
3. Insight 도출

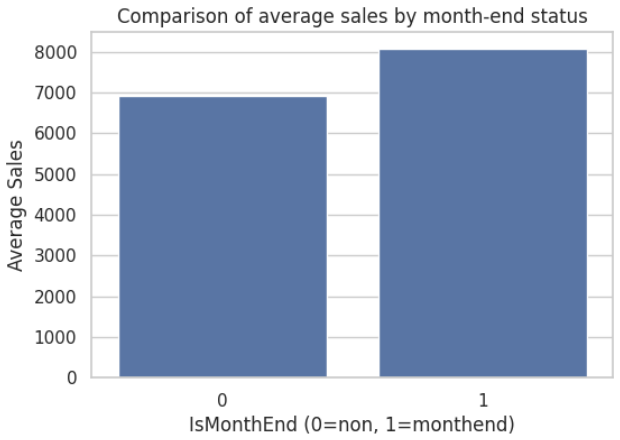
최종 Dataset의 컬럼별 Theme를 도출하여 각자 분석해보고 Theme별 Insight를 도출함

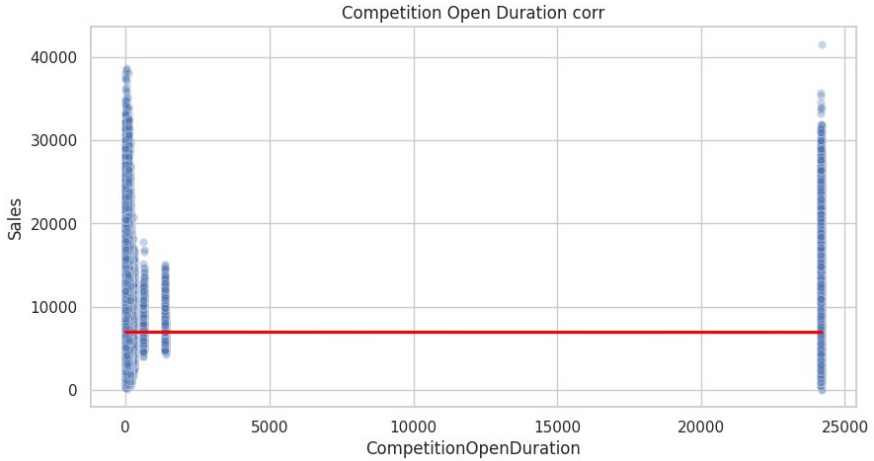
(4) 날짜 파생 변수 및 경쟁사/프로모션 기간 관련 파생 변수 기반







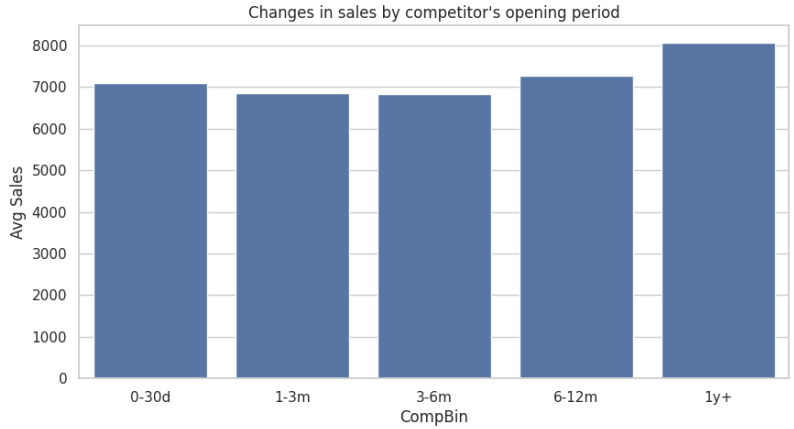


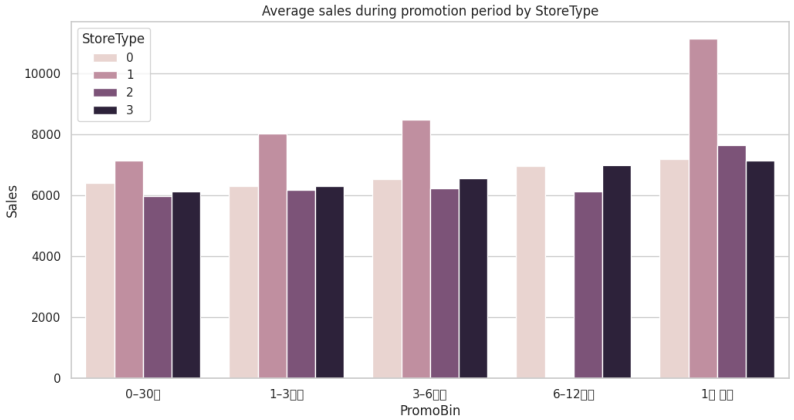


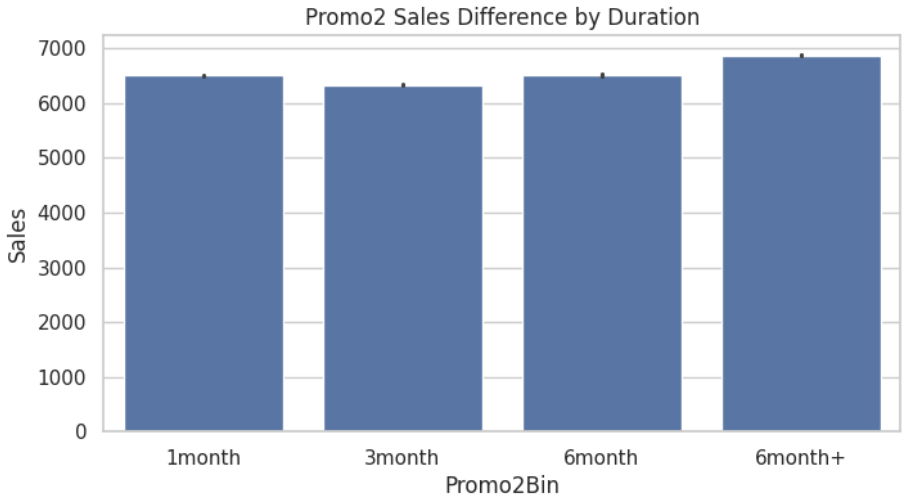
3. Insight 도출

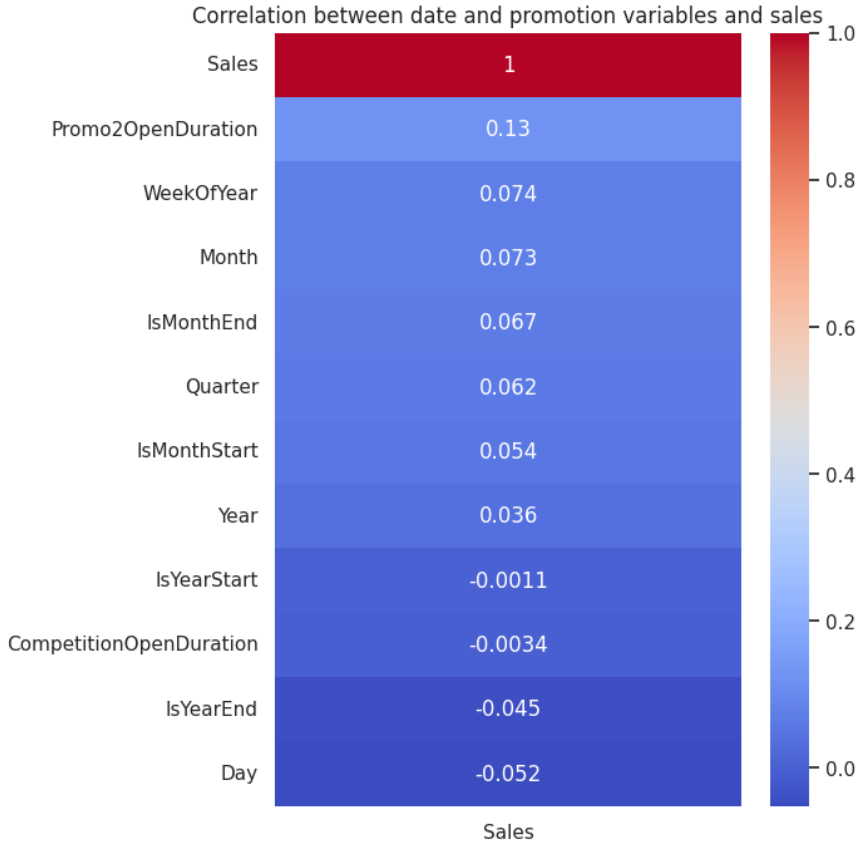
최종 Dataset의 컬럼별 Theme를 도출하여 각자 분석해보고 Theme별 Insight를 도출함

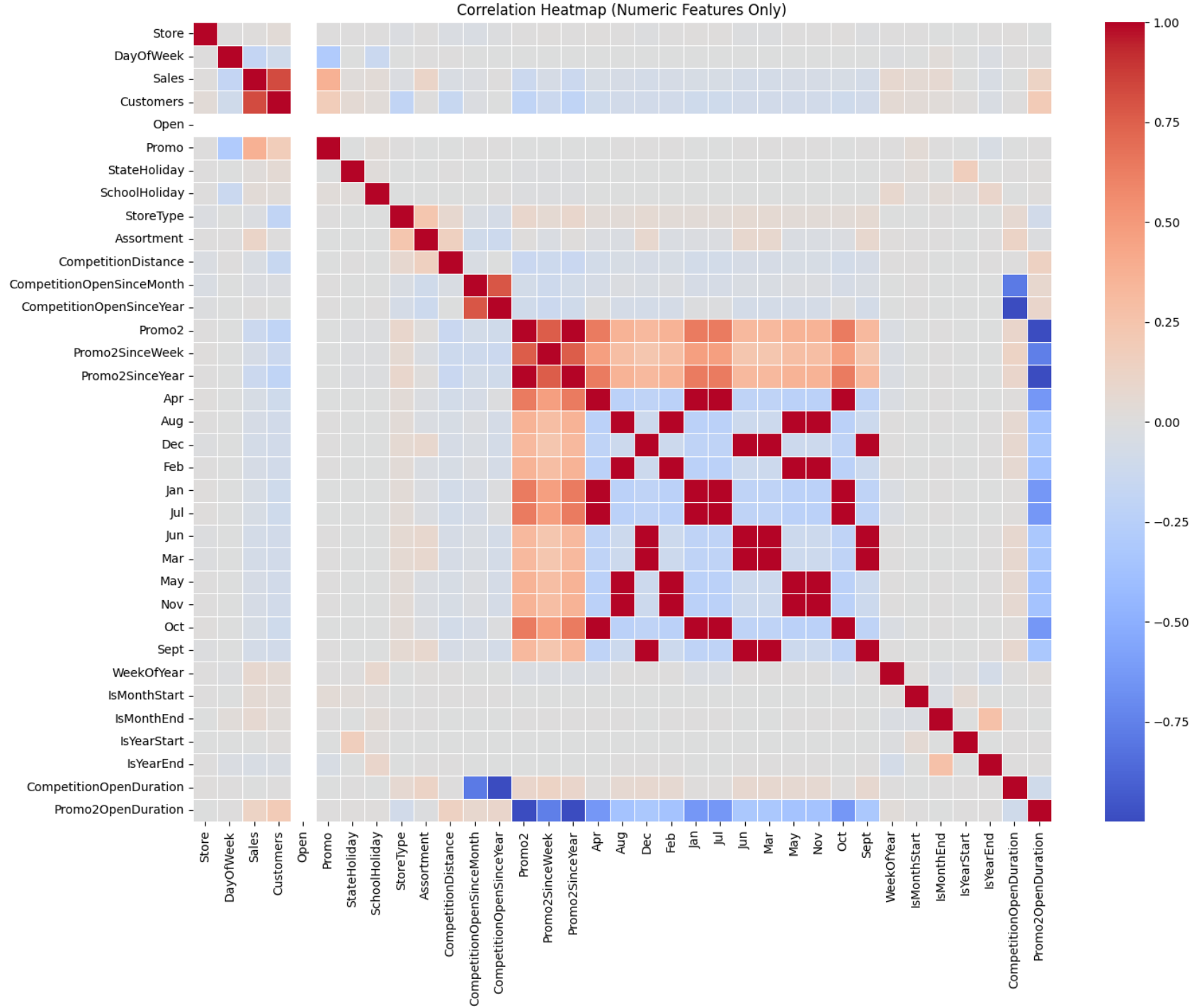
(4) 날짜 파생 변수 및 경쟁사/프로모션 기간 관련 파생 변수 기반










Correlation Heatmap (Numeric Features Only)
Store
DayOfWeek
Sales
Customers
Open
Promo
StateHoliday
SchoolHoliday
StoreType
Assortment
CompetitionDistance
CompetitionOpenSinceMonth
CompetitionOpenSinceYear
Promo2
Promo2SinceWeek
Promo2SinceYear
Apr
Aug
Dec
Feb
Jan
Jul
Jun
Mar
May
Nov
Oct
Sept
WeekOfYear
IsMonthStart
IsMonthEnd
IsYearStart
IsYearEnd
CompetitionOpenDuration
Promo2OpenDuration
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75

3. Insight 도출

인공지능 쓰지 않고 Feature 도출하기

Month(월) : *전반적으로 12월의 매출이 높은 편이었다.*

DayOfWeek(요일) : *전반적으로 일요일의 매출이 높은 편이었다.*

Cusmoters(고객 수) : *고객수가 높으면 매출도 높다.*

IsMonthEnd(월말 여부) : *주로 월말일 때 매출이 높았다.*

Promo(상시 프로모션) : *상시 프로모션을 하는 것이 시즌 프로모션보다 상대적으로 매출에 미치는 영향이 컸다.*

StateHoliday(공휴일) : *공휴일은 매출에 영향을 미친다. 크리스마스 시즌의 영향력이 가장 크다.*

StoreType(매장유형) : *B타입(슈퍼마켓 형태 대형 점포)이 상대적으로 매출이 높다.*

StoreType x Assortment(전시유형) : *B타입 점포의 Extended(프리미엄/전문 제품까지 포함하는 가장 큰 구성)의 매출이 높다.*

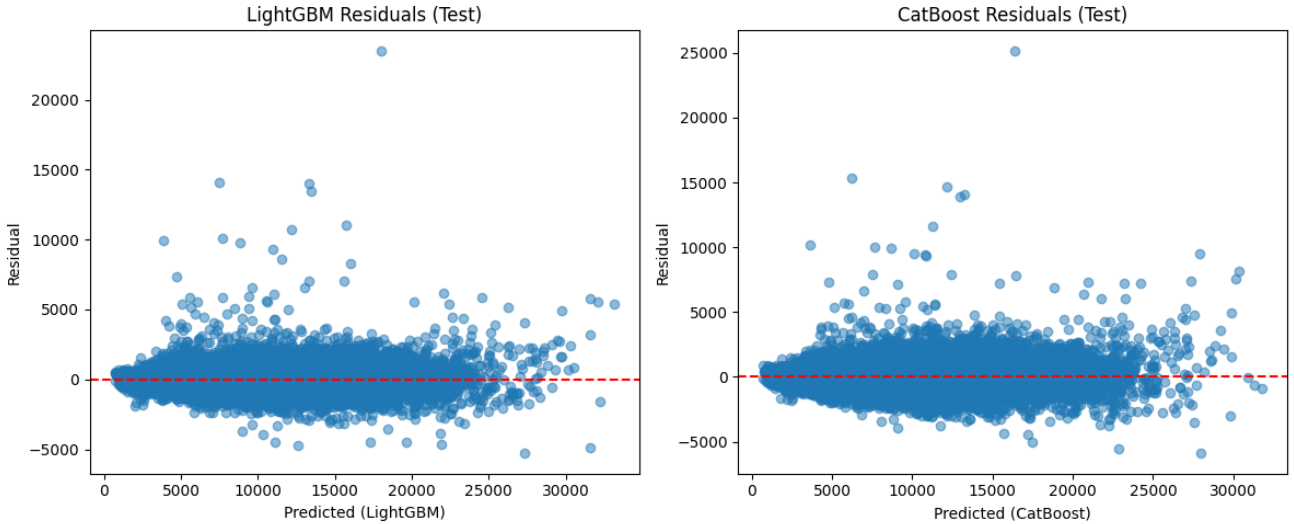
4. 최적 Model 도출

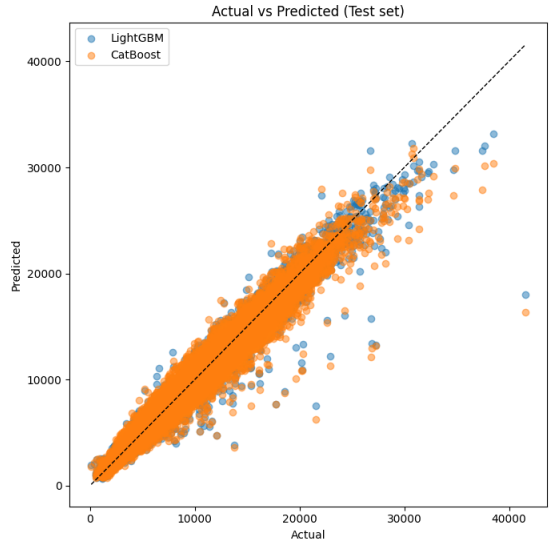
ML/DL 모델별 테스트를 수행하고 최적 모델을 도출함

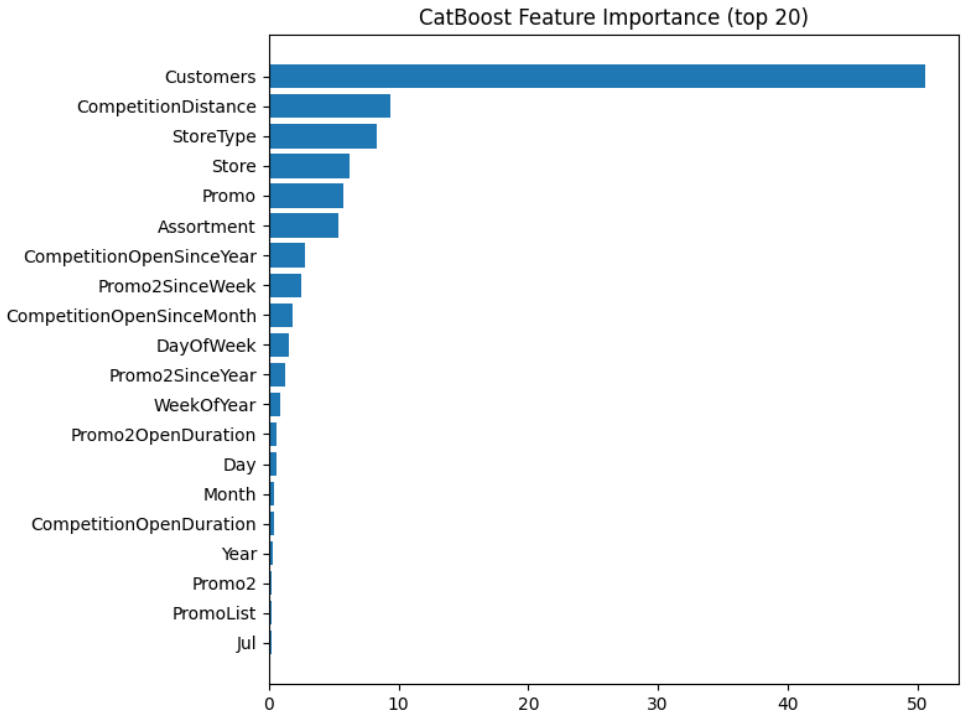
모델명	RMSE	MAE	R^2	비고
Random Forest	469.5302	312.5302	0.9771	Best Model
XGBoost	797.5473	578.3192	0.9340	
LightGBM	476.6559	332.0468	0.9764	
CatBoost	599.8146	421.2359	0.9626	
MLP	1,436.3629	1,007.9686	0.7853	DL모델
LSTM	740.7106	538.5830	0.9391	DL모델

4. 최적 Model 도출

모델별 시사점 요약-1 CatBoost & LightGBM

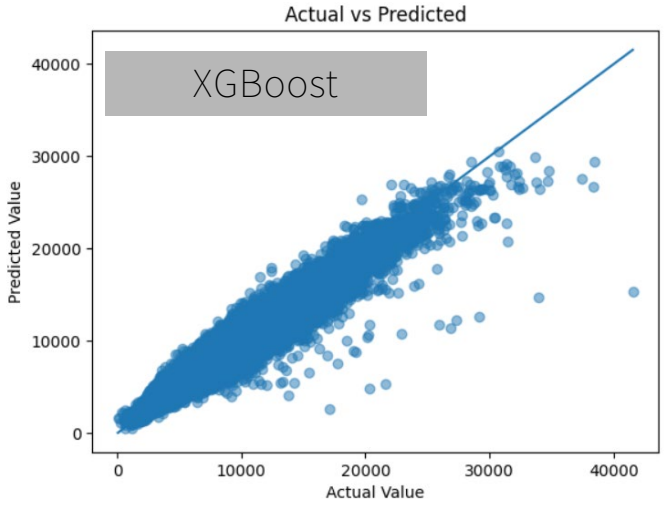


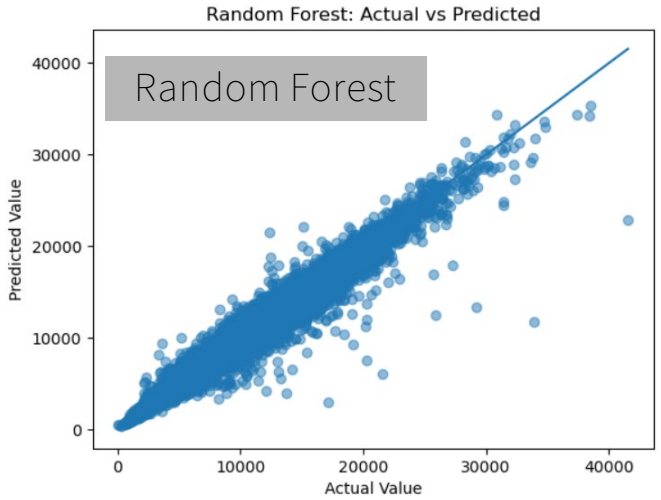


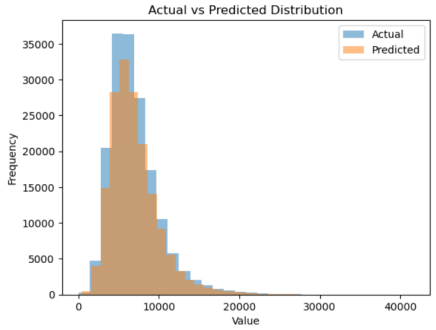


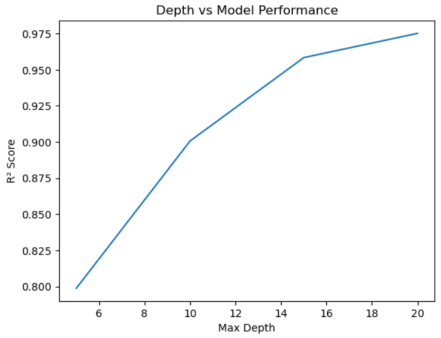
4. 최적 Model 도출

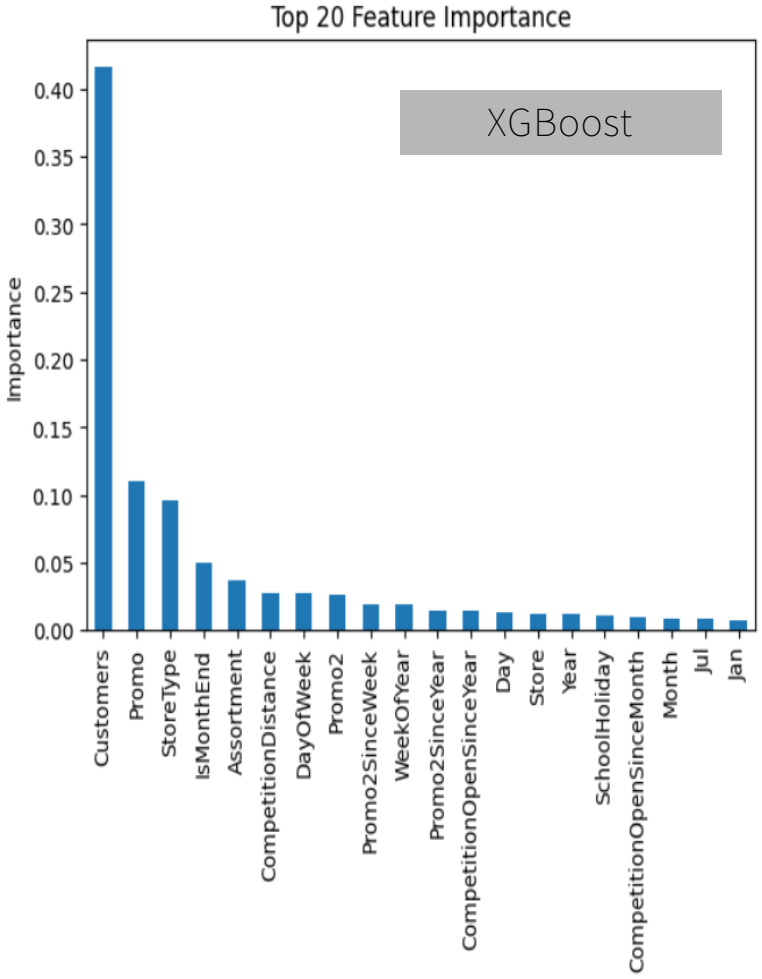
모델별 시사점 요약-2 XGBoost & Random Forest

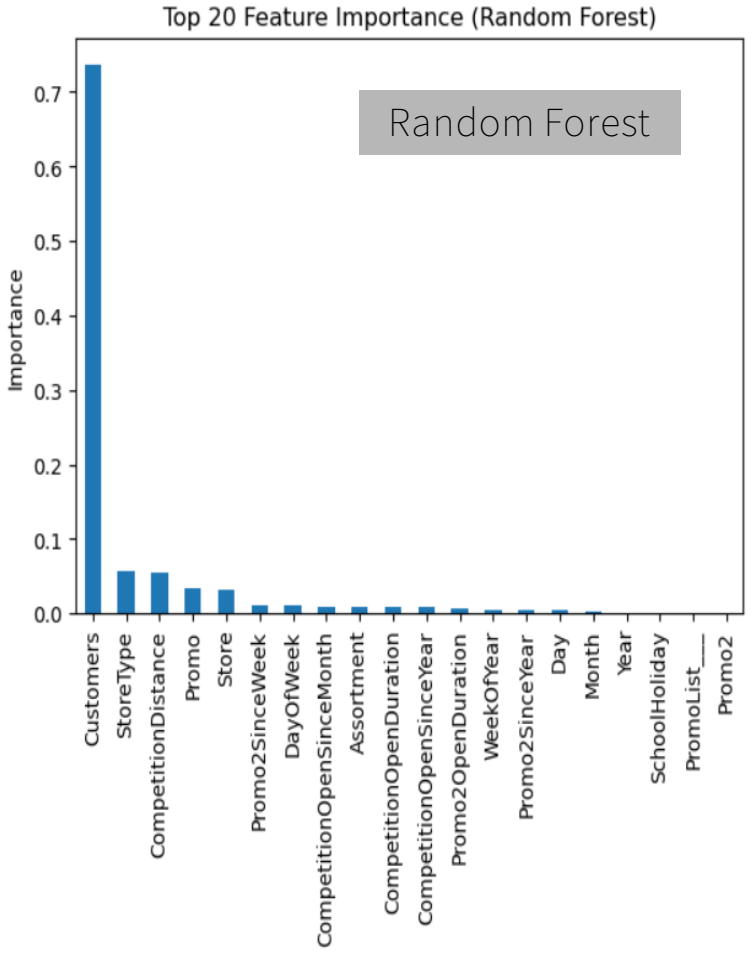






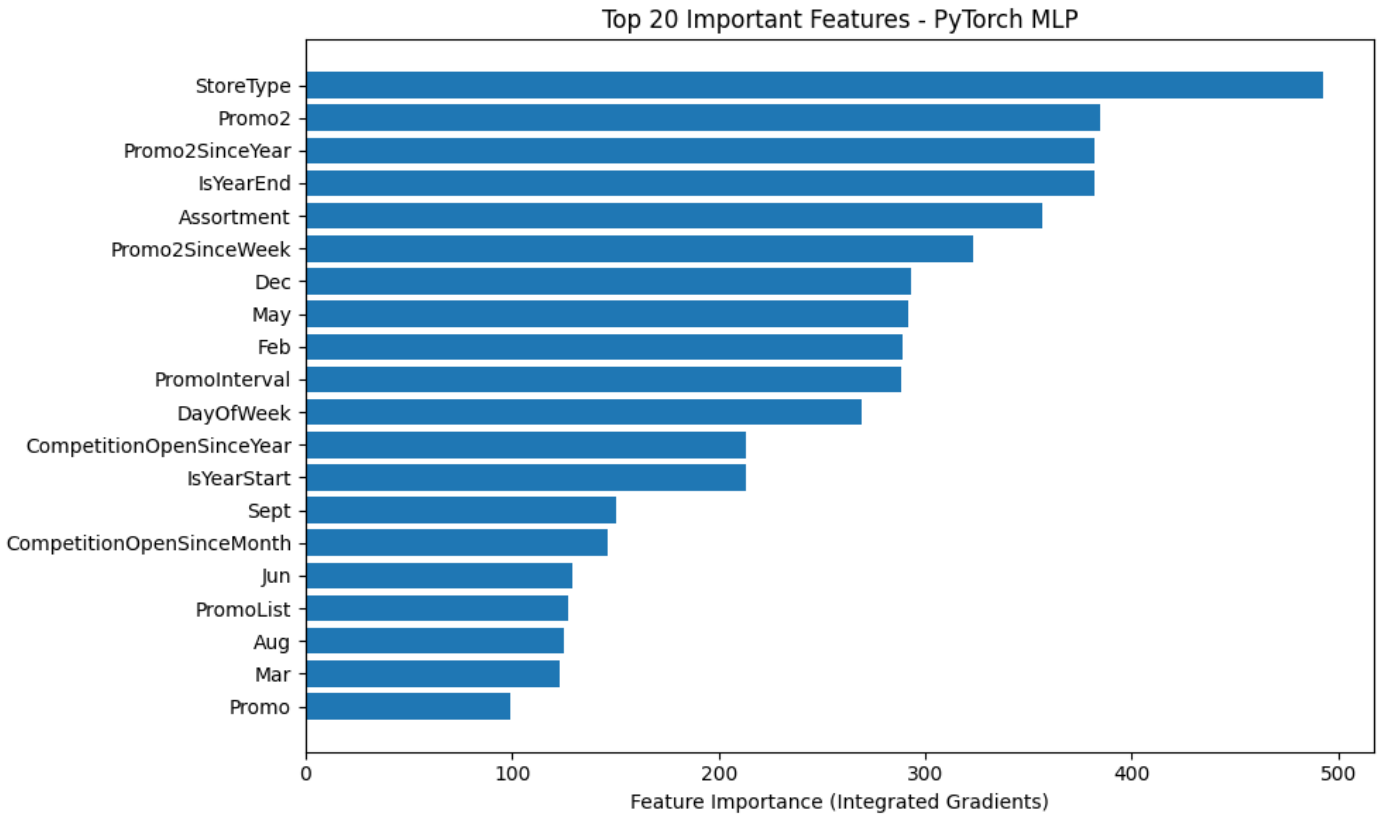


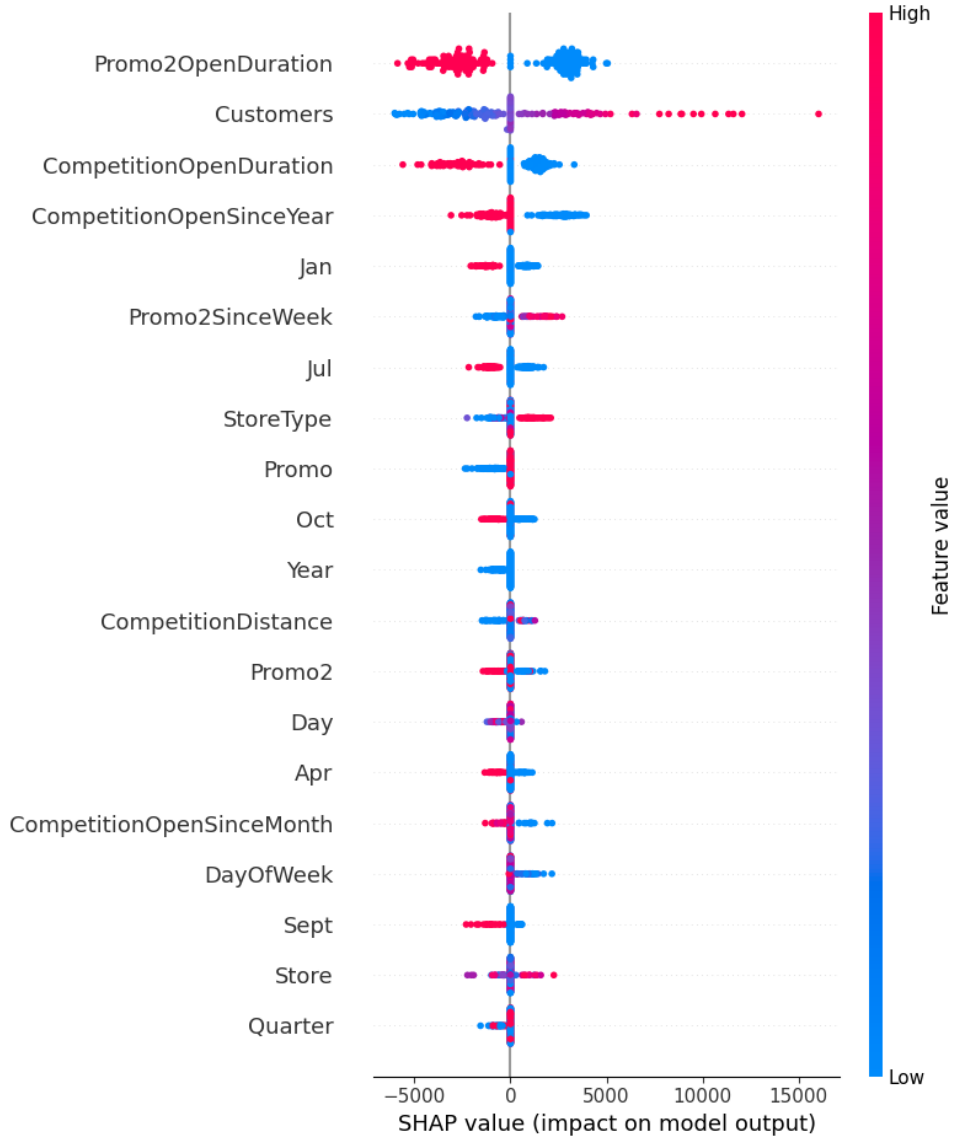




4. 최적 Model 도출

모델별 시사점 요약-3 MLP / LSTM





5. 모델별 가상 시나리오 분석

각 모델별 상위 Feature를 바탕으로 가상 시나리오를 세워서 교차분석을 수행함

1. 경쟁자의 거리(CompetitionDistance)가 가깝거나 멀면 어떻게 되는가?

2. 상시 프로모션(promo)을 하거나 하지 않으면 어떻게 되는가?

3. Storetype이 바뀌면 매출에 영향을 미치는가?

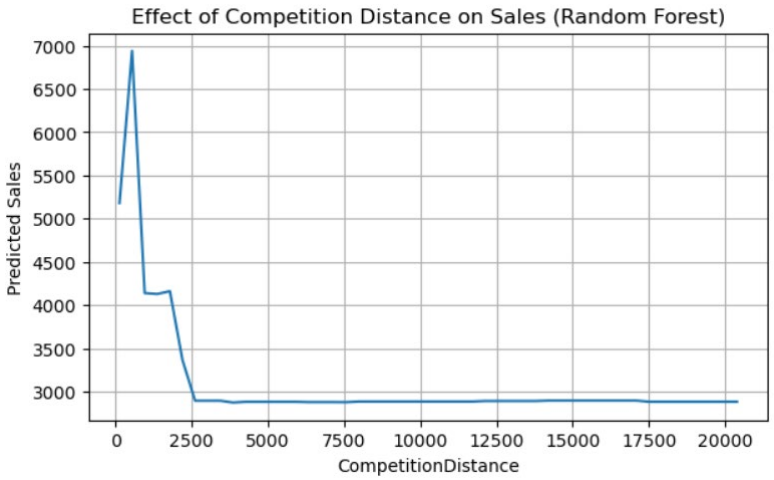
4. 시즌 프로모션(promo2)의 지속기간 장단이 매출에 영향을 미치는가?

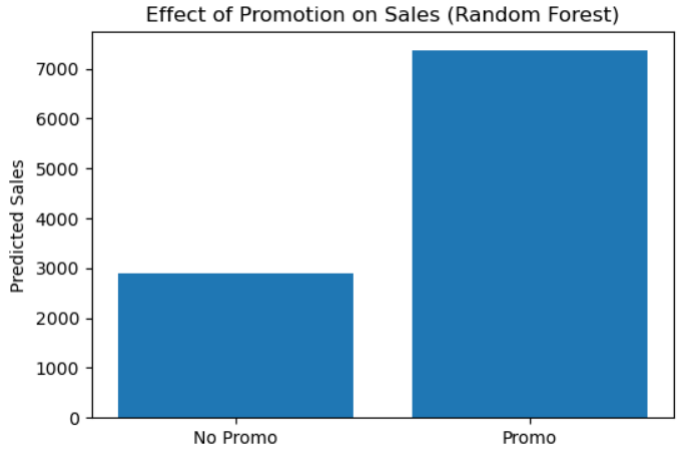
5. 경쟁자가 길게 살아남을수록(CompetitionOpenDuration) 우리 매장 매출에 영향을 미치는가?

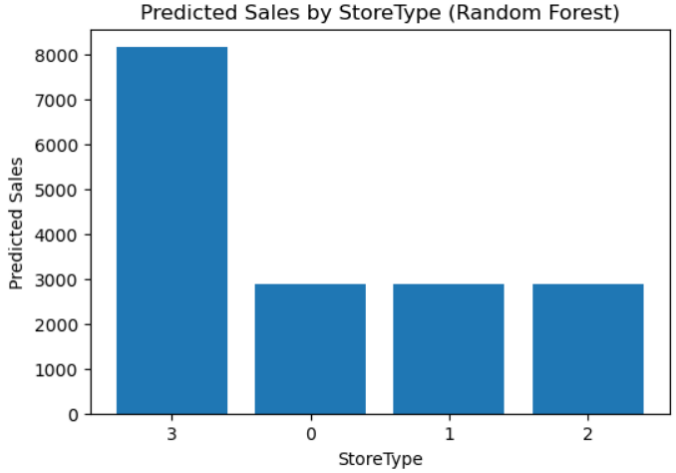
6. A type이나 B type 매장(Storetype)의 전시유형(Assortment)을 바꾸면 매출에 영향을 미치는가?

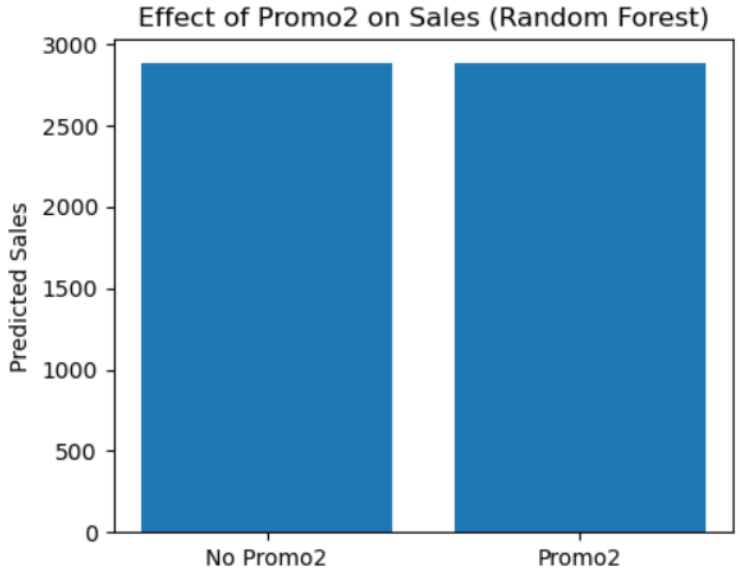
5. 모델별 가상 시나리오 분석

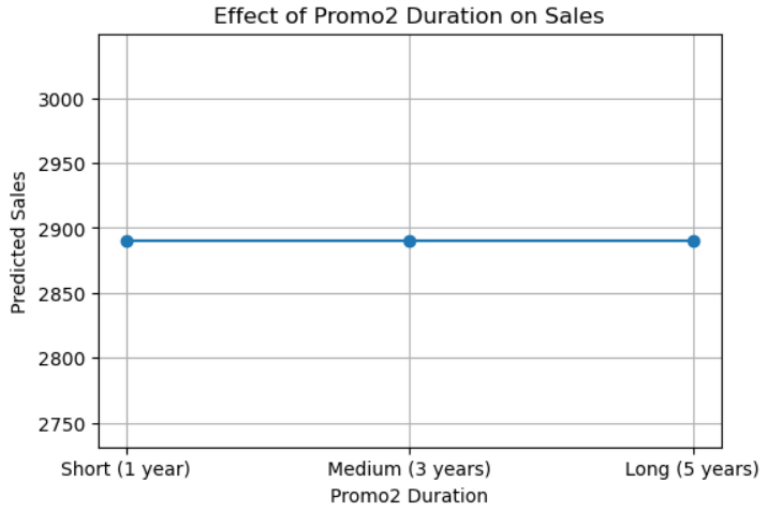
1. Random Forest

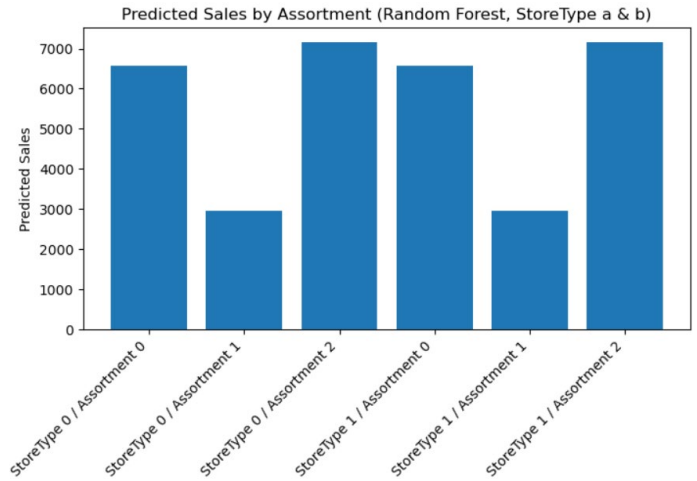






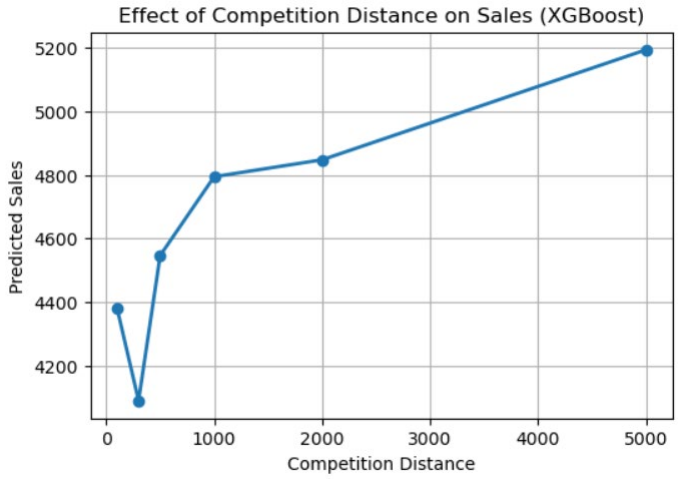


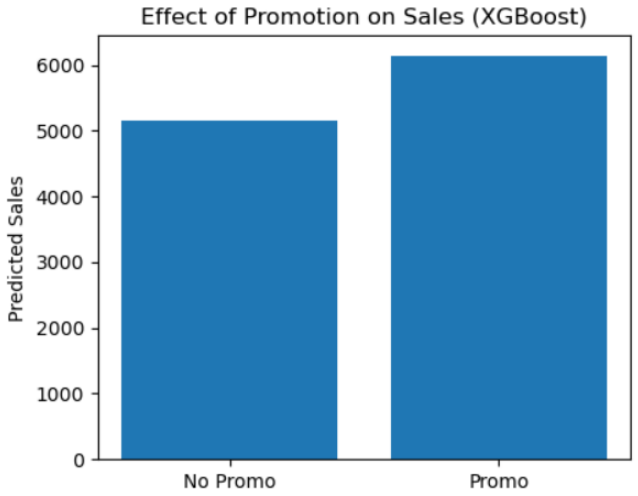


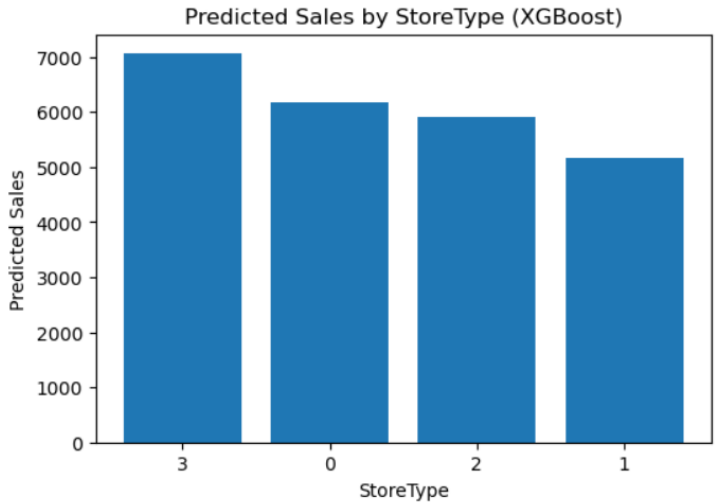


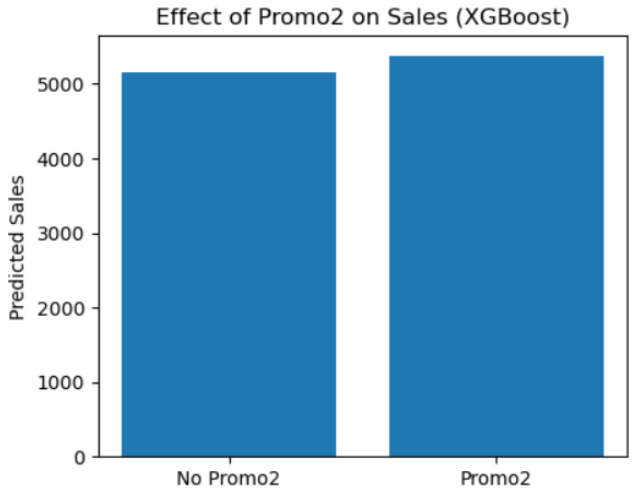
5. 모델별 가상 시나리오 분석

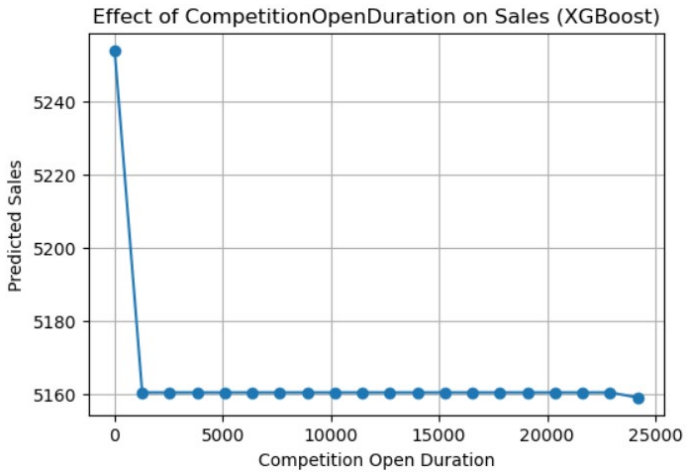
2. XGBoost

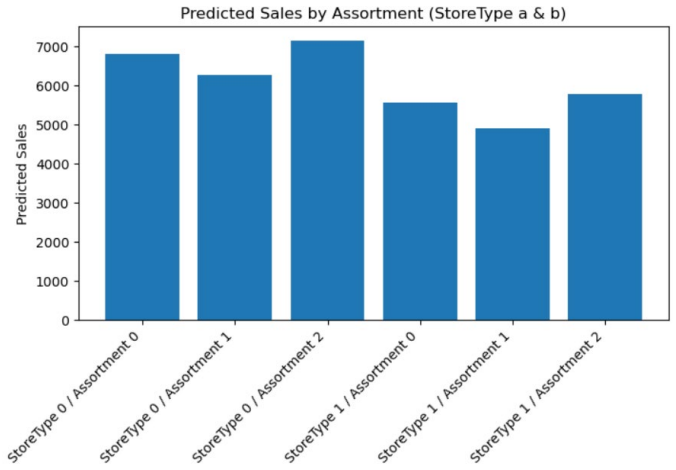






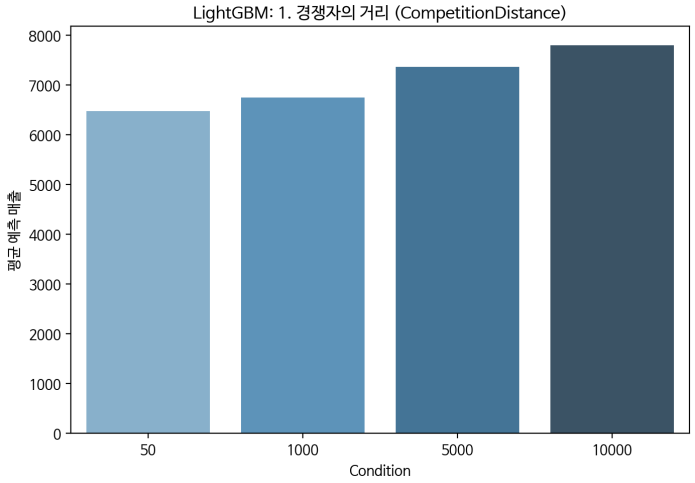


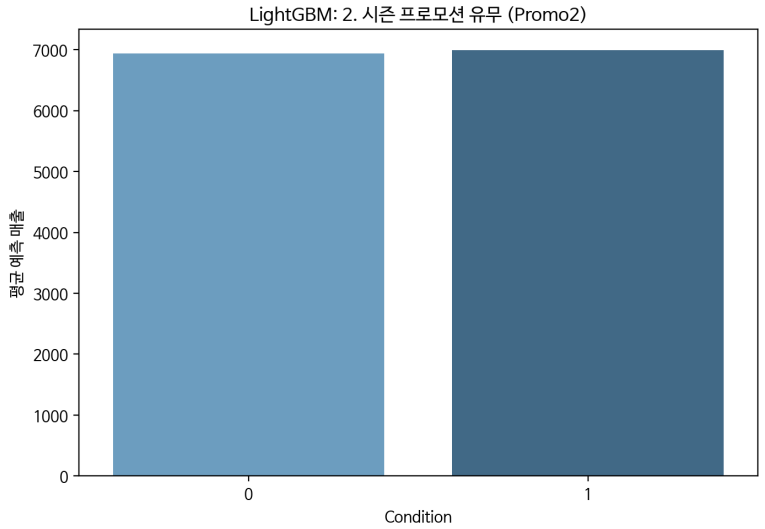


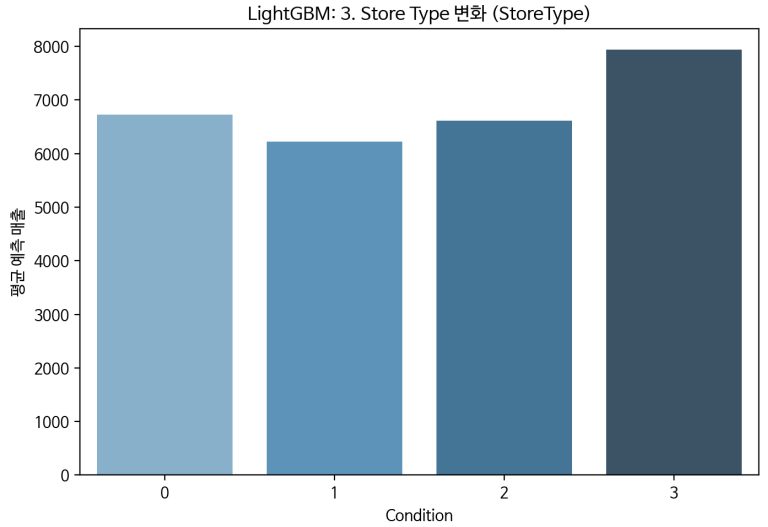


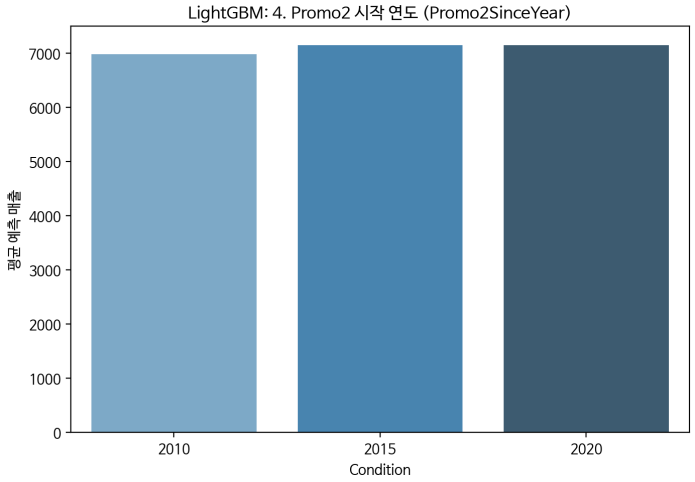
5. 모델별 가상 시나리오 분석

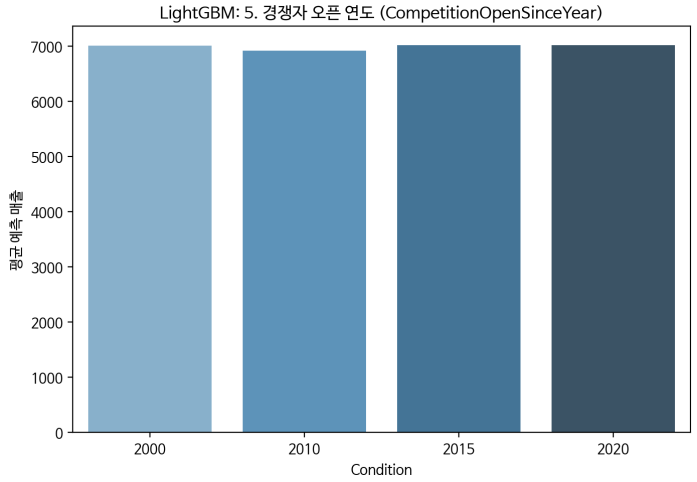
3. LightGBM

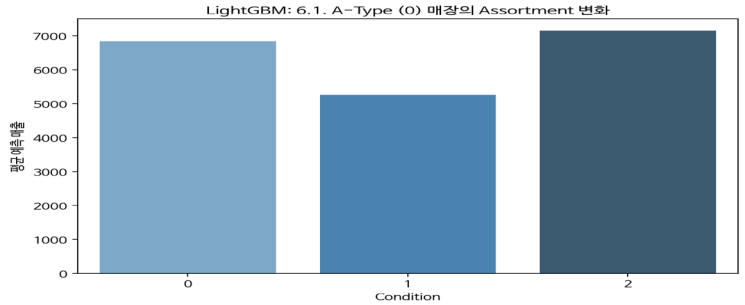


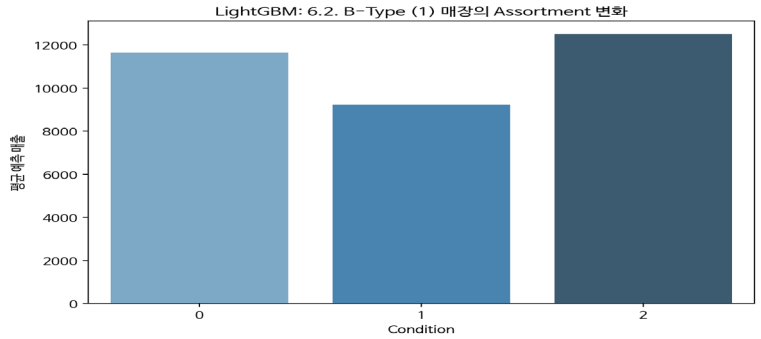






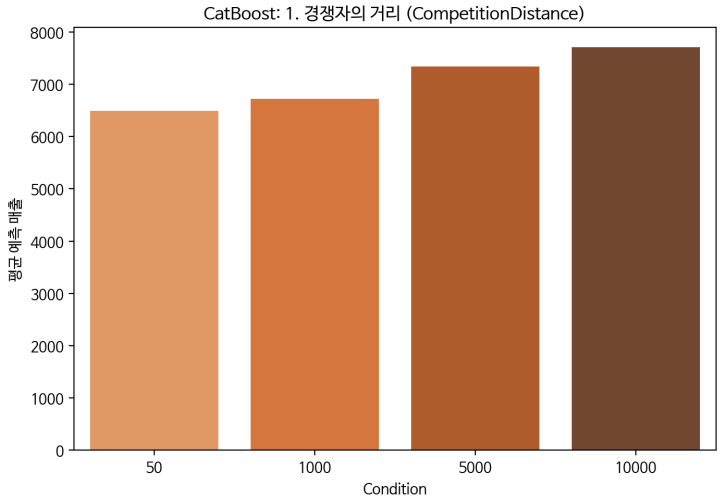


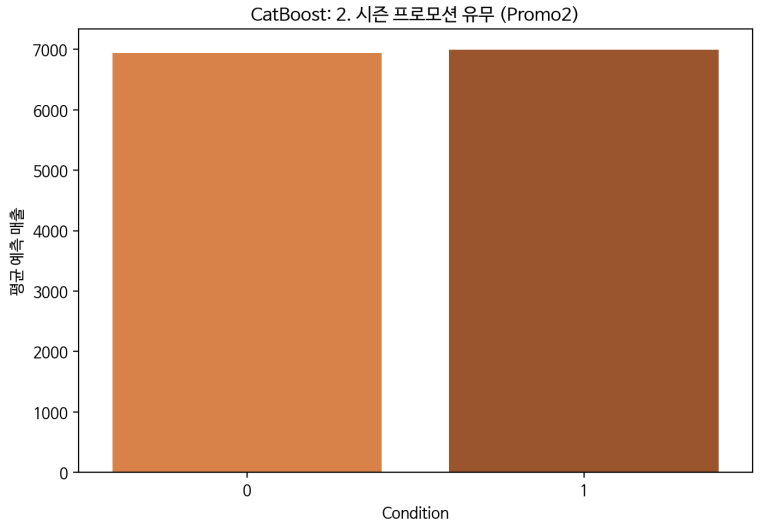


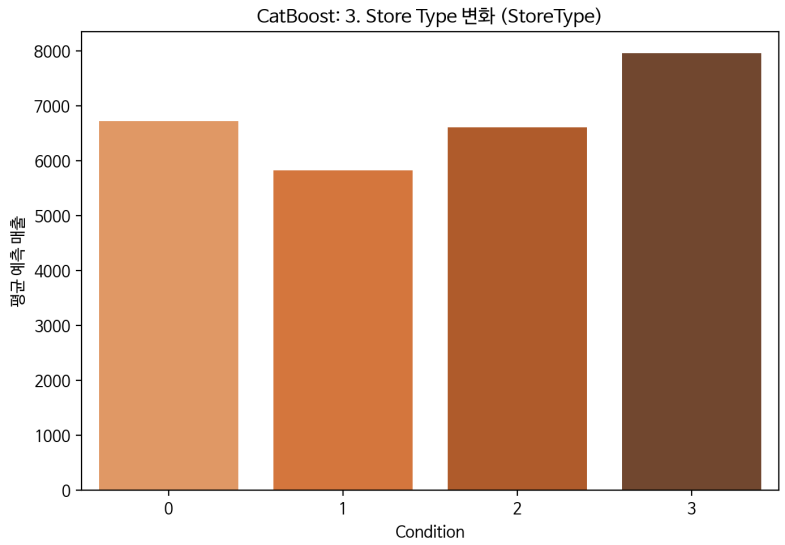


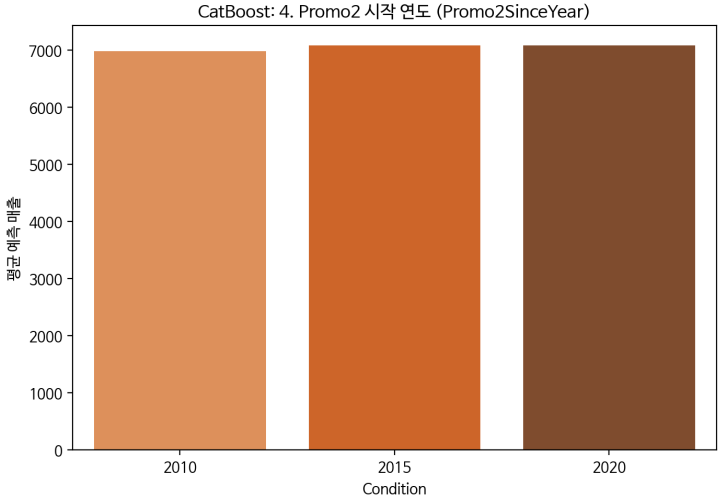
5. 모델별 가상 시나리오 분석

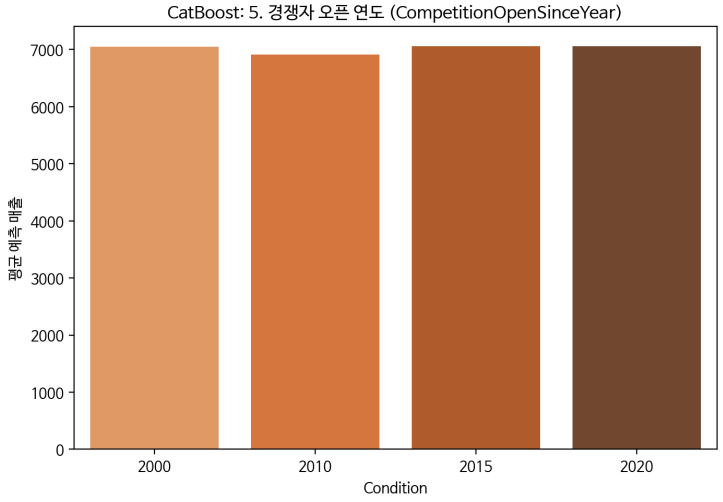
4. CatBoost

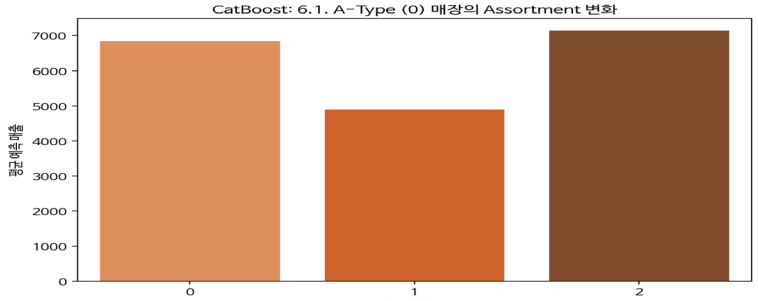


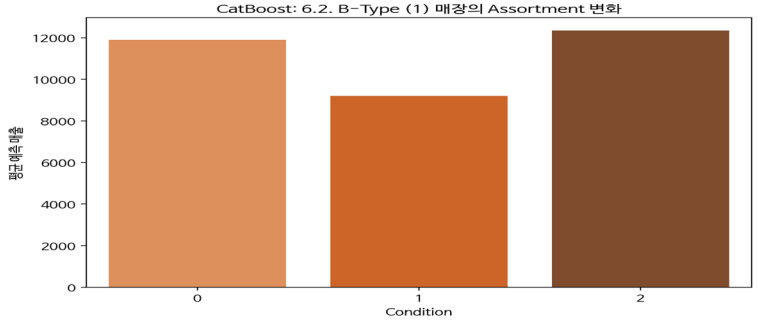






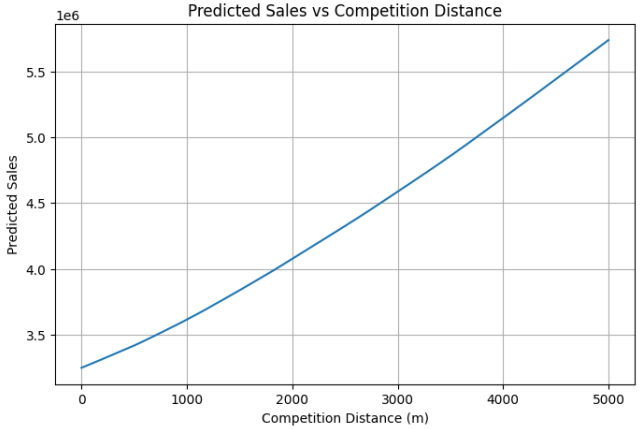


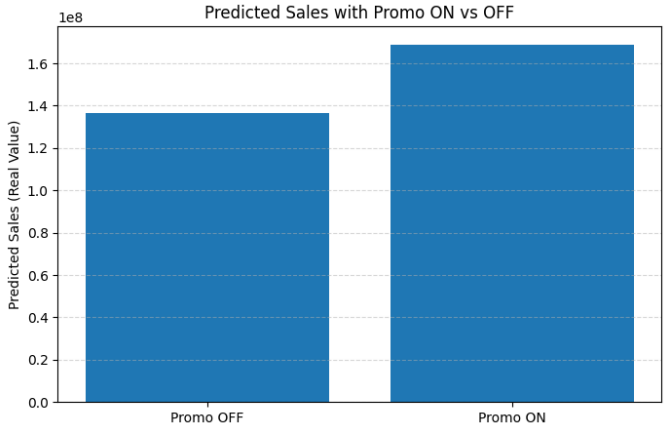


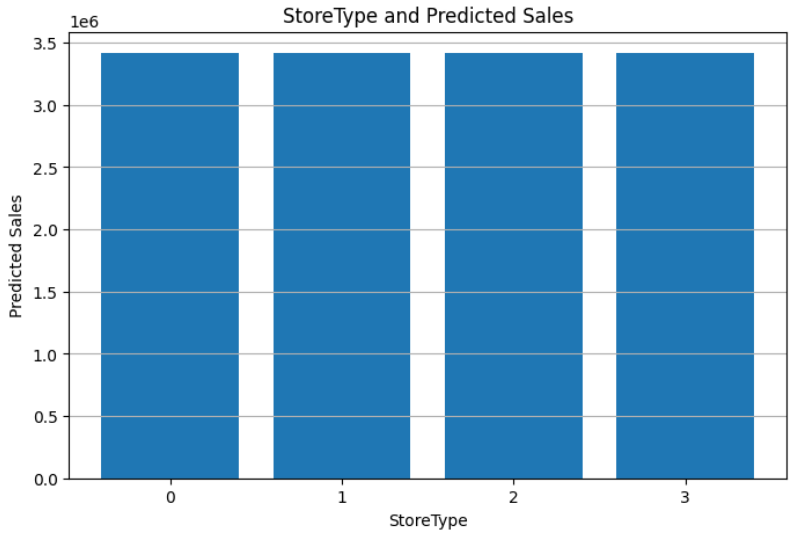


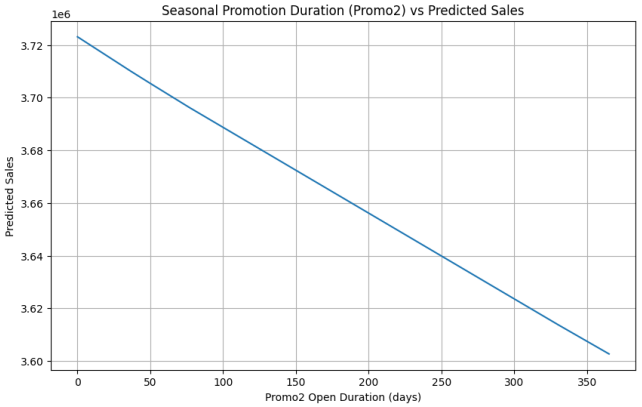
5. 모델별 가상 시나리오 분석

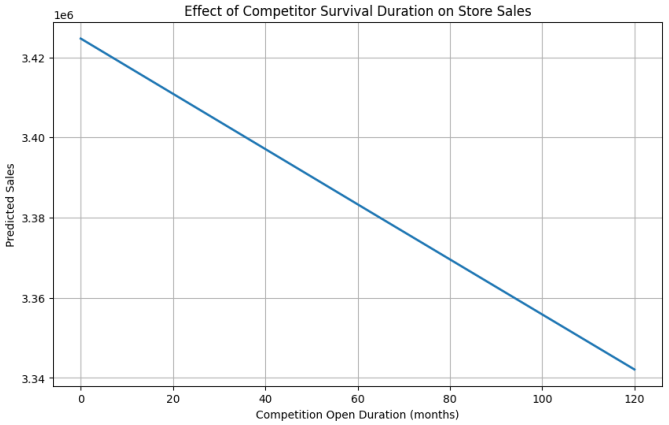
5. MLP

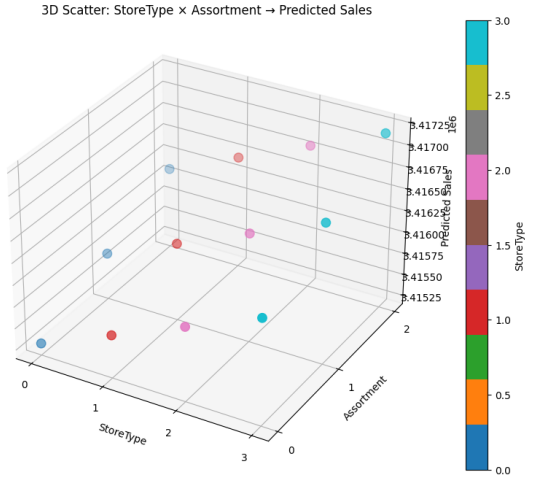






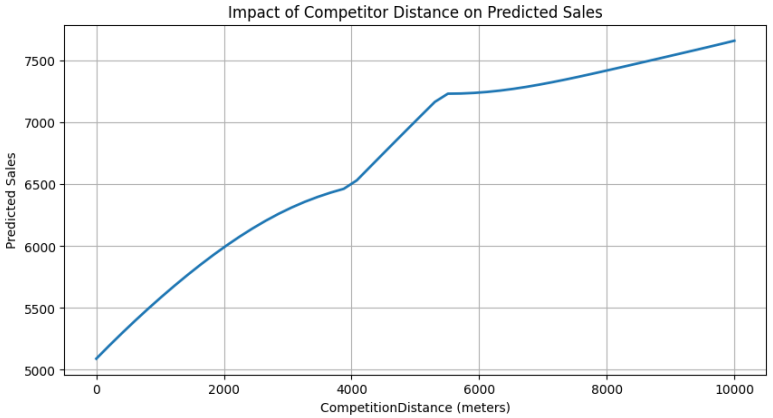


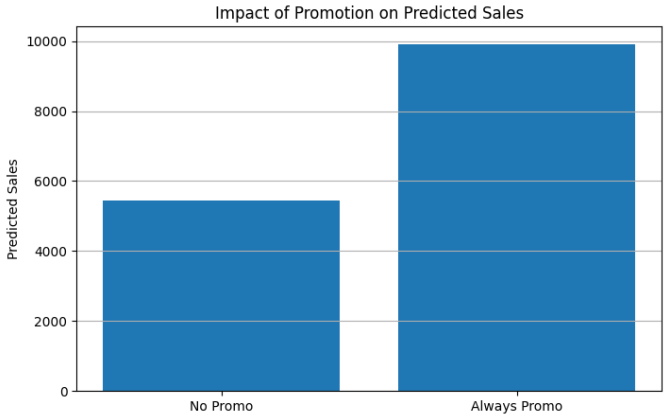


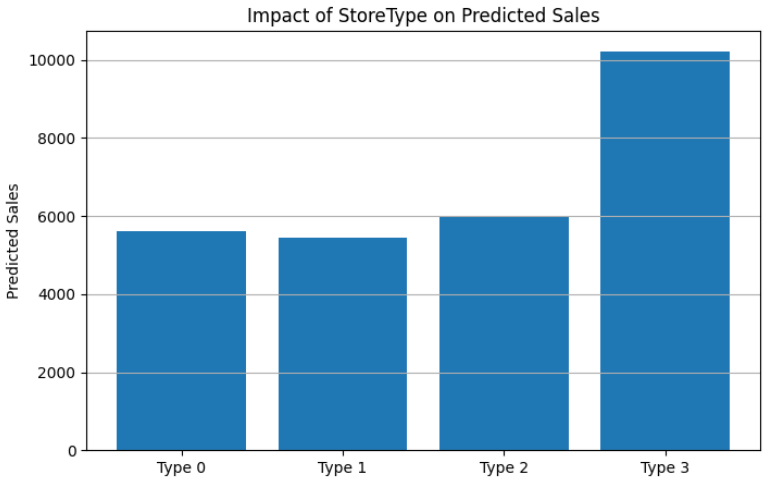


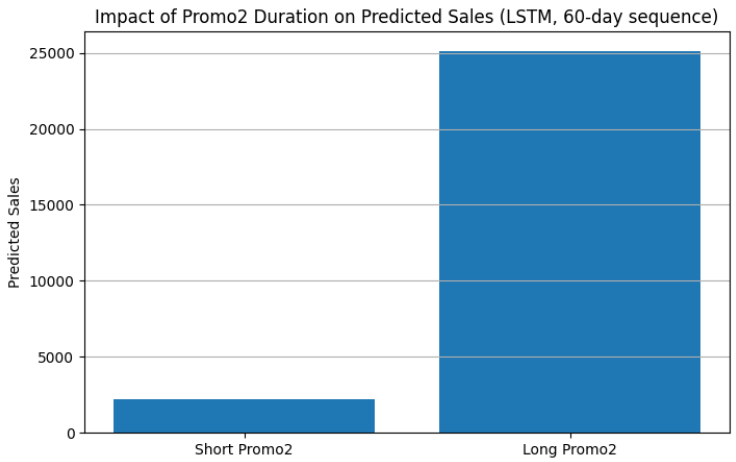
5. 모델별 가상 시나리오 분석

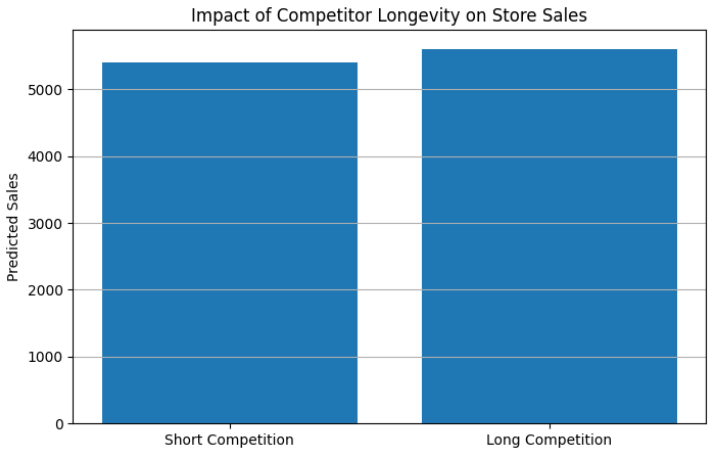
6. LSTM

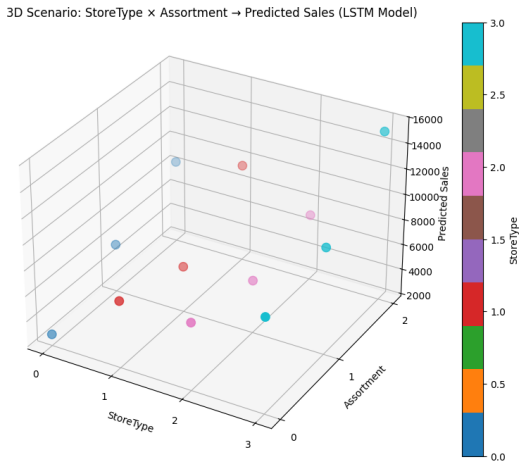












6. 매출 향상 방안

1. Promotion 최적화 전략

Promo(상시 프로모션)

- 경쟁사 프로모션 회피형 Calendar 운영
- 경쟁사 프로모션 시점을 예측/수집하고 상이한 주차에 Promo 실행
- 실제 문서 시나리오 분석에서 “경쟁사와 겹치면 효과 감소”가 확인됨

- Promo 강도 차등화
- 고객이 적은 월(2~3월 등)에는 Promo 집중 배치
- 매출이 이미 높은 월(12월 등)은 필수일 때만 최소 운영

- 요일·월말 효과 기반의 Promo 집중 전략
- DayOfWeek, IsMonthEnd가 매출 상승과 연관된 근거 존재
- 월말 + 특정 요일에 타겟 프로모션을 집중

Promo2 (시즌제 캠페인 형태)

- 매출이 낮은 월(2월, 3월 등)을 타겟으로 Promo2 집중 배치

- StoreType × Assortment 조합별로 Promo2 효과를 A/B 테스트

- 경쟁 환경이 안정적인 지역 (CompetitionDistance 멀고 Duration 길음)에만 운영
- 종합 heatmap에서도 Duration 증가 시 영향 미미한 것으로 나타남

6. 매출 향상 방안

2. StoreType & Assortment 기반 '매장 포트폴리오 최적화'

- 잠재력 높은 A·C 타입 매장에 Assortment 확장 적용 실험
- 시나리오 분석에서도 "Assortment 변경에 따라 매출 차이 존재"가 확인
- 특히 A타입 점포에 "Extended 구성" 시험적 적용 → 매출 상승 가능성

- StoreType별 전시 정책 차별화
- B타입은 전문·프리미엄 라인업 비중 확대
- C·D타입은 "핵심 카테고리 집중형" 구성으로 효율 강화

- 지역별 경쟁 거리(CompetitionDistance) 기반로 StoreType 재배치
- 경쟁사와의 거리(CompetitionDistance)는 '매출 향상 유무'와 연결시키기 보다, 경쟁사와의 거리가 가까운 곳들의 Storetype을 전략적으로 배치하는 방식으로 연결짓기
- 경쟁이 심한 지역(A타입) → B타입 수준의 구성 강화
- 경쟁이 약한 지역 → 기본형으로도 충분 (비용 절감)

7. 한계와 아쉬운 점

- 결측치 처리를 했을때 open 여부가 0인 경우를 모두 제외하고 학습을 시켰는데, 모두 1로 바꾸고 학습을 시켰을 경우 결과가 달라질 수도 있다는 것을 나중에 깨달았음
- 경쟁사와의 거리가 가까울 경우 경쟁사가 얼마나 오래 체류했는지에 따라 매출이 바뀌는지도 확인했어야 하는데 시간관계상 생략됨
- 전반적으로 분석할 때 customer를 빼거나 정규화를 하고 분석했으면 다른 결과가 나올지에 대해서도 의문이 생김
- 좀 더 풍부한 분석(SHAP 등)이나 Dummy Data Test 등을 해보고 싶었는데 리소스가 제한과 정해진 시간 등으로 못 해본 것들이 많음
- 모델을 돌리고 나서는 가중치와 모델을 저장하는 습관을 들이자