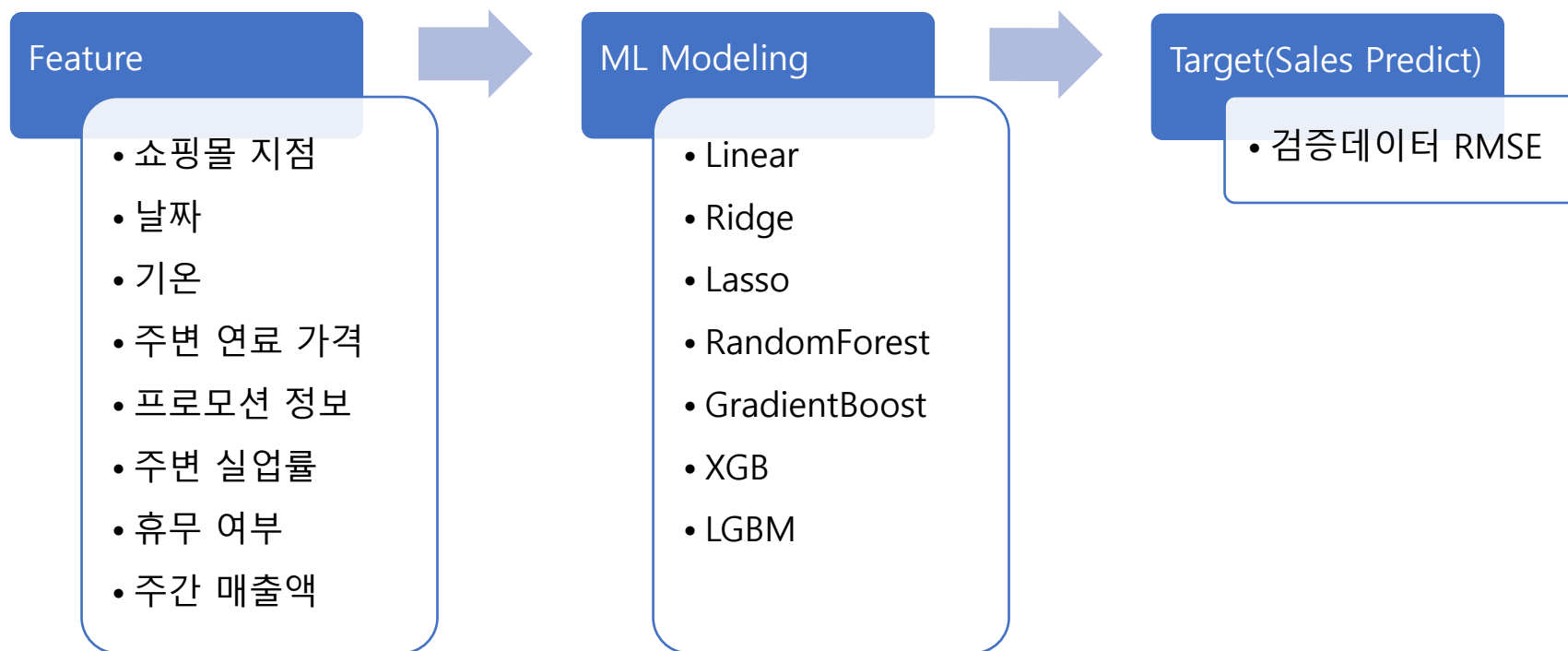


1. 프로젝트 개요 및 설계

프로젝트 목표

쇼핑몰의 프로모션, 주변 기온, 휴일 정보 등을 이용한 지점별 한 달 매출액 예측

모델 기본 설계



2. 진행과정

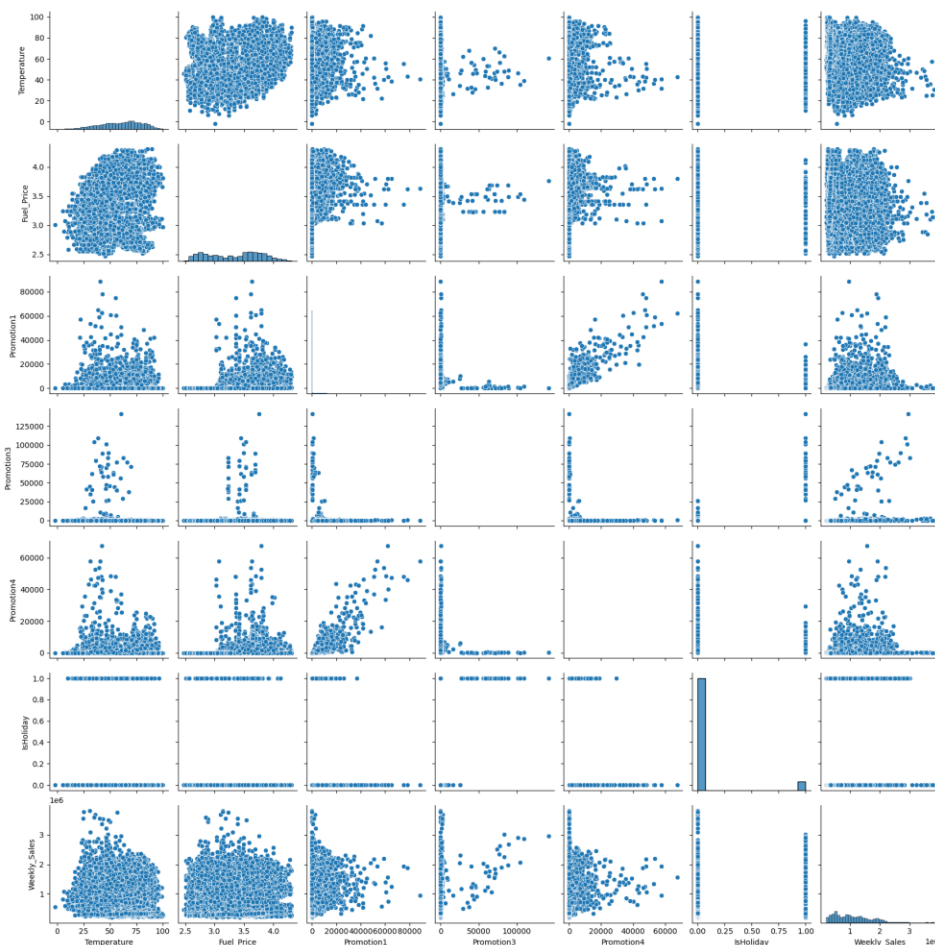
쇼핑몰 지점별 매출액 예측 |

EDA

전처리

Modeling

실행결과



- Plotly, Matplotlib, Seaborn 사용
- 주변요인 직접적이거나 관련성이 높은지 우선 점검
- 실업률, 기온, 연료비, 프로모션 직접 관련성 낮음
- 날짜와 휴일, 지점 위주 모델링 결정

2. 진행과정

쇼핑몰 지점별 매출액 예측 | EDA 전처리 Modeling 실행결과

```
# 결측값 처리
train = train.fillna(0)
test = test.fillna(0)

# 날짜 Feature 생성 (일, 월, 연도)
# '일/월/연' 형식을 '연-월-일' 형식으로 변환
train['Date'] = pd.to_datetime(train['Date'], format="%d/%m/%Y").dt.strftime("%Y-%m-%d")

# 다시 datetime 형식으로 변환하여 연, 월, 일 추출
train['Date'] = pd.to_datetime(train['Date'])
train['Year'] = train['Date'].dt.year # 연(year)
train['Month'] = train['Date'].dt.month # 월(month)
train['Day'] = train['Date'].dt.day # 일(day)

# '일/월/연' 형식을 '연-월-일' 형식으로 변환
test['Date'] = pd.to_datetime(test['Date'], format="%d/%m/%Y").dt.strftime("%Y-%m-%d")

# 다시 datetime 형식으로 변환하여 연, 월, 일 추출
test['Date'] = pd.to_datetime(test['Date'])
test['Year'] = test['Date'].dt.year # 연(year)
test['Month'] = test['Date'].dt.month # 월(month)
test['Day'] = test['Date'].dt.day # 일(day)

# 휴일 Feature 변환 (True/False -> 1/0)
train['IsHoliday'] = train['IsHoliday'].astype(int)
test['IsHoliday'] = test['IsHoliday'].astype(int)
```

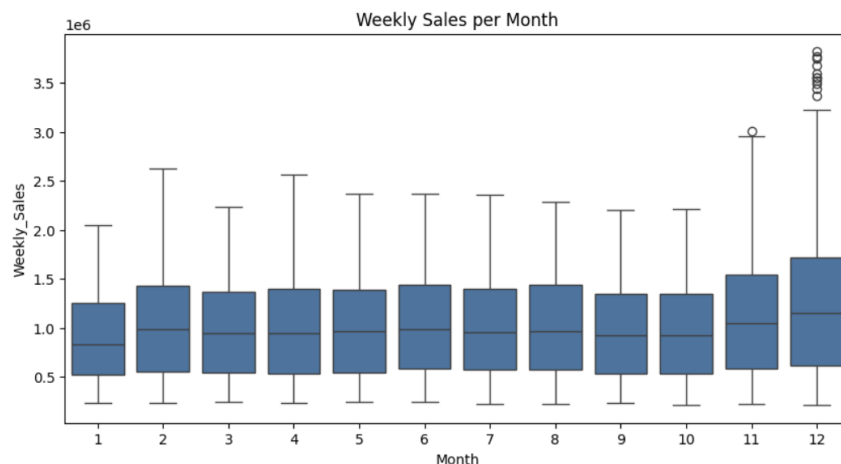
```
# 불필요 칼럼 삭제
train=train.drop(columns=['id'])
train=train.drop("Temperature",axis=1)
train=train.drop("Fuel_Price",axis=1)
train=train.drop(['Promotion1','Promotion2','Promotion3','Promotion4','Promotion5'],axis=1)
train=train.drop("Unemployment",axis=1)

test=test.drop(columns=['id'])
test=test.drop("Temperature",axis=1)
test=test.drop("Fuel_Price",axis=1)
test=test.drop(['Promotion1','Promotion2','Promotion3','Promotion4','Promotion5'],axis=1)
test=test.drop("Unemployment",axis=1)
```

○ 날짜 형식 피쳐 엔지니어링

○ 불필요 칼럼 삭제

○ 엔지니어링 후 시각화



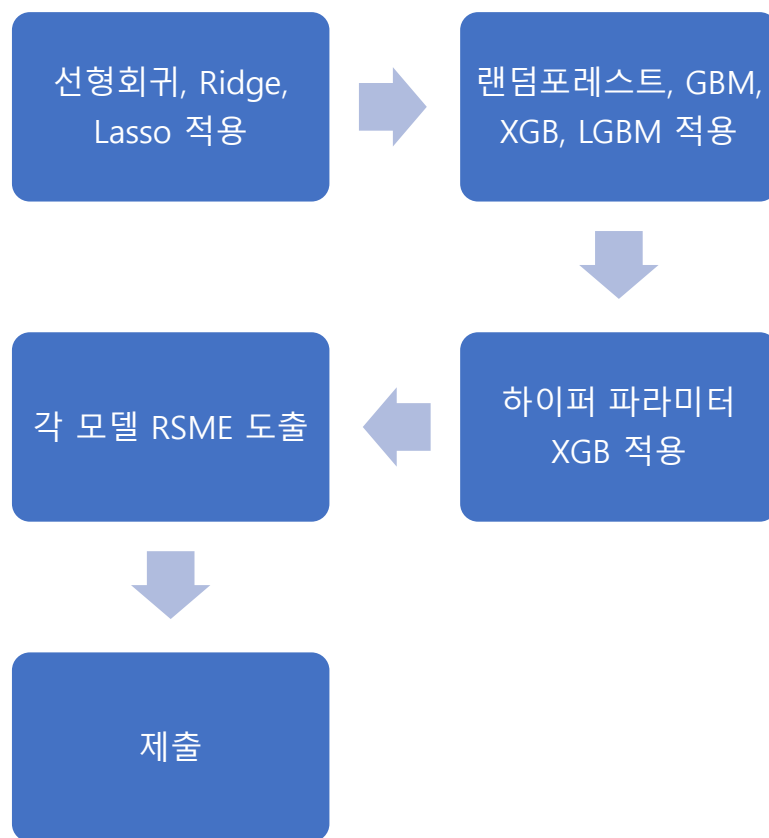
2. 진행과정

쇼핑몰 지점별 매출액 예측 | EDA

전처리

Modeling

실행결과



- 회귀 모델별 적용
- 하이퍼 파라미터 조정작업
- 오차 값 도출

2. 진행과정

쇼핑몰 지점별 매출액 예측 | EDA 전처리 Modeling **실행결과**

최신순		점수순		
	제목	제출 일시	public점수 private점수	제출선택
1131759	submission.csv edit	2025-02-11 11:37:41	474747.8716538172 465483.6625826616	<input checked="" type="radio"/>
1132976	250212_submission.csv pythonize submission edit	2025-02-12 12:36:42	434682.9732177729 433942.6019565688	<input type="radio"/>
1139205	250218_submission_XGB_Tuning.csv XGBoost Hyper Parameter Tuning edit	2025-02-18 10:17:02	47481.9557062611 43255.9662126128	<input type="radio"/>
1138155	250217_submission_XGB.csv xgb model applied edit	2025-02-17 14:45:14	43051.7089306773 47710.247779589	<input type="radio"/>
1135127	250214_submission_RF.csv random forest applied edit	2025-02-14 13:07:49	40808.020256527 40134.7214190146	<input type="radio"/>

- 모델링 결과와 달리 RandomForest 결과가 가장 높은 점수 획득
RandomForestRegressor : 40,808
XGBoost : 43,051
- 하이퍼 파라미터 등 적용에 의해 train 데이터 과적합 원인
- 모델 순위 상위 4.2% 달성