

# 요일, 월, 계절 파생변수 추가

## 월별 판매량 합친 파생변수

```
date_columns = train_data.columns[4:] # 날짜 컬럼들 선택
seasonal_data = train_data.copy()

# 날짜 수만큼 반복
for col in date_columns:

    if col >= '2022-01-01' and col <= '2023-04-04':

        #2022-01까지만 추출
        year_month = col[:7]

        #선언 안되었으면 선언
        if year_month not in seasonal_data.columns:
            seasonal_data[year_month] = 0

        #2022-01에 1월달 값 다 넣기
        seasonal_data[year_month] += train_data[col]
        #이후 반복
```

이 코드를 통해 열 인덱스의 앞부분만 가지고와서 월별로 카운트함.

## 요일별 판매량 합친 파생변수

날짜 인덱스들만 datetime형식으로 바뀌어서 dayofweek 메소드를 통해 요일별로 합산함.

```
# 열 인덱스들을 날짜 형식으로 변환
date_columns = train_data.columns[4:] # 날짜 컬럼들 선택
date_columns = pd.to_datetime(date_columns)

# 열 이름을 변경하여 날짜 형식으로 열을 추가
weekday_data = train_data.copy()
weekday_data.columns = train_data.columns[:4].tolist() + date_columns.tolist() # 날짜 형식으로 변경

# 요일별 합산 열 추가
dayofweek_columns = date_columns.dayofweek # 요일(0: 월요일, 6: 일요일)
weekday_names = ['월요일', '화요일', '수요일', '목요일', '금요일', '토요일', '일요일']
for i, day in enumerate(weekday_names):
    seasonal_data[day] = weekday_data[date_columns[dayofweek_columns == i]].sum(axis=1)

# 변환된 데이터프레임 출력
print(seasonal_data)
```

## 계절별 판매량 합친 파생변수

```
# 계절별 월평균 열 추가, 2023-04는 뺐음
seasonal_data['봄'] = seasonal_data[['2022-03', '2022-04', '2022-05', '2023-03']].mean(axis=1)
seasonal_data['여름'] = seasonal_data[['2022-06', '2022-07', '2022-08']].mean(axis=1)
```

```
seasonal_data['가을'] = seasonal_data[['2022-09', '2022-10', '2022-11']].mean(axis=1)
seasonal_data['겨울'] = seasonal_data[['2022-01', '2022-02', '2022-12', '2023-01', '2023-02']].me
```

2023-04는 데이터가 적어서 12,1,2월은 겨울, 3,4,5월은 봄, 6,7,8월은 여름, 9,10,11은 가을로 함.