데이콘 Basic
“쇼핑몰 지점별 매출액 예측 경진대회”

쇼핑몰 지점별 매출액 예측

* 대회 설명

쇼핑몰 매장별 프로모션 정보, 주변 날씨, 실업률, 연료 가격 등의 정보를
바탕으로 쇼핑몰의 각 지점별 주간 매출액을 예측
https://www.dacon.io/competitions/official/235942/overview/description

* 변수

id : 샘플 아이디
Store : 쇼핑몰 지점
Date : 주 단위(Weekly) 날짜
Temperature : 해당 쇼핑몰 주변 기온
Fuel_Price : 해당 쇼핑몰 주변 연료 가격
Promotion 1~5 : 해당 쇼핑몰의 비식별화 된 프로모션 정보
Unemployment : 해당 쇼핑몰 지역의 실업률
IsHoliday : 해당 기간의 공휴일 포함 여부
Weekly_Sales : 주간 매출액 (목표 예측값)

* 가설

날짜 관련 요인 : Date + Temperature + IsHoliday
경제적 요인 : Fuel_Price + Unemployment
기타 요인 : Promotion 1~5
날짜 관련 요인이 기본적으로 매출에 영향을 주고 경제적 요인이나 기타 요인에
의해서도 매출에 영향을 줄 것

쇼핑몰 지점별 매출액 예측

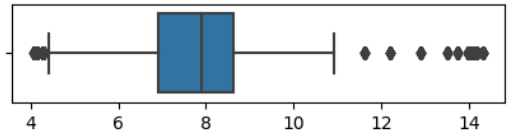
EDA & 전처리

1

```
RangeIndex: 6255 entries, 0 to 6254
Promotion1    2102 non-null
Promotion2    1592 non-null
Promotion3    1885 non-null
Promotion4    1819 non-null
Promotion5    2115 non-null
```

Promotion1 ~ 5 결측값이 너무 많음 → 결측값과 음수를 0으로 대체

2



이상값 제거

3

```
IsHoliday     6255 non-null   bool
```

Bool 타입 int형으로 변환

4

```
train['Date'] =  pd.to_datetime(train['Date'], format = '%d/%m/%Y')
train['Year'] = train['Date'].dt.year
train['Month'] = train['Date'].dt.month
train['Week'] = train['Date'].dt.isocalendar().week
train['Day'] = train['Date'].dt.day
```

Date의 object 형태를 datetime형태로 바꾸고 Year, Month, Week, Day로 새로운 변수 생성

5 # 이번 대회의 고득점 방법 ! (고득점자 코드에서 따온 것)

```
time_series_df = pd.DataFrame()
for num in range(1, 46) :
    col_name = "Store " + str(num)
    # test 셋은 2012년 10월 데이터이기 때문에 각 연도별 10월(포함) 이전의 시계열 데이터의 유사성을 판단하여 type을 나누었습니다.
    time_series = train[(train.Store==num) & (train.Month <= 10)]['Weekly_Sales'].values
    time_series_df[col_name] = time_series
transpose_time_series_df = time_series_df.transpose()
```

타겟 변수를 기준으로

```
kmeans = TimeSeriesKMeans(n_clusters = 3, metric='dtw', max_iter = 300, init='k-means++', random_state = 0).fit(transpose_time_series_df)
transpose_time_series_df['cluster'] = kmeans.labels_
average_score = silhouette_score(transpose_time_series_df, transpose_time_series_df['cluster'])
```

군집을 나누어서

```
for i in range(len(prediction)) :
    if prediction[i] == 0 :
        list_0.append(i+1)
    elif prediction[i] == 1 :
        list_1.append(i+1)
    else:
        list_2.append(i+1)
```

```
for i in range(len(prediction)) :
    if prediction[i] == 0 :
        train.loc[(train.Store == i + 1), 'Type'] = 0
        test.loc[(test.Store == i + 1), 'Type'] = 0
    elif prediction[i] == 1 :
        train.loc[(train.Store == i + 1), 'Type'] = 1
        test.loc[(test.Store == i + 1), 'Type'] = 1
    else:
        train.loc[(train.Store == i + 1), 'Type'] = 2
        test.loc[(test.Store == i + 1), 'Type'] = 2
```

파생변수 생성

6

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

예측에 도움 안되는 변수 제거

쇼핑몰 지점별 매출액 예측

* 모델링

	RMSE
xgboost	0.0520
catboost	0.0548
lightgbm	0.0642
rf	0.0655

좋은 성능을 낸
학습 알고리즘
중 4개를 스태킹

	RMSE
Mean	0.0496

* 인사이트

모든 변수가 영향을 줄 거라고 생각했지만 날짜 관련 요인 중에서도 기온을 뺀 시계열 데이터만 넣고 예측했을 때 가장 큰 성능이 나타난다는 것을 알아냄 → 이 쇼핑몰들의 매출액은 날짜에 가장 큰 영향을 받음

* 배운 점

- 일부 변수만 가지고 (feature selection) 예측을 할 때 더 높은 예측력이 나타날 수도 있다는 것을 배움
- 시계열 군집 tslearn을 배움
- 타겟 변수를 기준으로 군집을 나누어서 파생변수를 만드는 것을 배움