OSSP 프로젝트 2차 진행보고서

OSSP_3P_1

2013113097 산업시스템공학과 정순우

2018111749 의생명공학과 손민영

2016111671 경영학부 문지용

- 목차 -

*** 현재 단계 : 데이터 모델링 & 시각화 (11/18 ~ 11/23)**

세부 과정	주차별 추진계획				단계별 성과
	1주차	*2주차*	*3주차*	*4주차*	
데이터 수집 & 전처리					CSV 파일 정리
EDA & 모델링					모델링
모델링 결과 평가 & 시각화					통계적 수치, 그래프
프로젝트 결과 보고서 작성					보고서

- 데이터 모델링 단계
- 모델링 결과 시각화 단계

I. 3주차 진행 상황

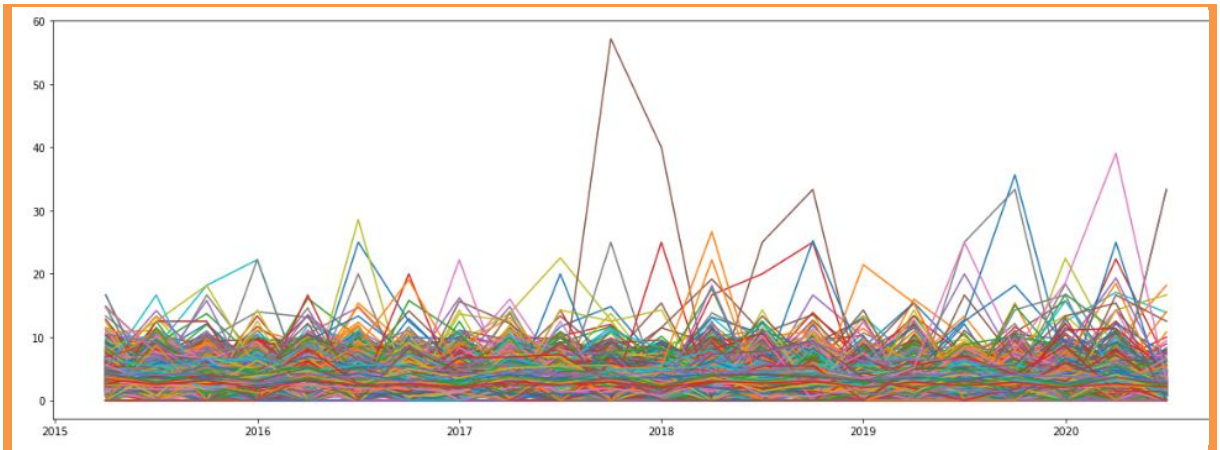
1. 모델링

- 각자 모델을 하나씩 선정하여 모델링 과정 진행

2. Xgboost 모델 (Extreme Gradient Boosting)

(1) 모델 선정 근거

1496 개의 상권의 폐업률을 2015년 1분기~2020년 2분기 기간별로 시각화를 하면 결과가 아래와 같다.



< 서울 1496개 상권별 폐업률 >

위 시각화 결과로 알 수 있는점은 다음과 같다.

- 대다수 상권의 폐업률이 0~10% 수준에 해당하며, 일부 상권의 폐업률은 20%까지 상승한다. 그리고 낮은 확률로 30%를 넘는 폐업률을 보여준다.

- 대다수의 상권은 폐업률이 항상 높은 상태로 유지되지 않는다. 특정 년도, 분기에 폐업률이 높았다면 다음 분기에 감소하는 패턴을 보인다.

일부 상권들과 대다수의 상권들의 차이를 설명할 수 있는 모델을 선정하려 했다. 또한 분석 결과로 폐업률의 원인이 되는 요소를 찾을 수 있는 모델을 선정하려 했다.

Xgboost 의 경우 다음과 같은 장점이 있다.

- Feature 중요도를 분석 결과로써 제공한다. (n번 의사결정 트리를 생성하며, 매번 Feature weight를 갱신하는 Xgboost 특유의 알고리즘 때문이다.)

- Training을 하며 교차 검증이 가능하다. (저번 2차 중간결과 발표시 피드백 받았던, 머신러닝 결과가 신뢰할 수 있는지에 대해서 모든 데이터로 n번 교차검증하는 기능을 지원하여 머신러닝 모델이 학습데이터에 대해서 Overfit 되는 문제를 방지하고 Genereal 한 문제에 대해서 성능을 높일 수 있다.)

- 회귀문제 및 분류 문제 모두를 지원한다.

위 이유들로 Xgboost 머신러닝 모델을 선정하였다.

(2) 모델 기능 정의

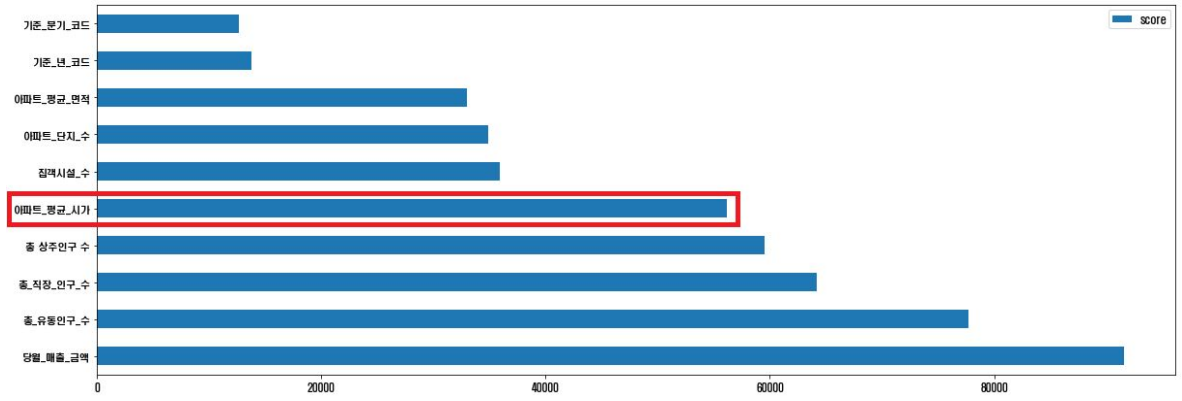
(3) 현재 진행 단계

학습한 데이터 : 2015년 1분기 ~ 2020년 2분기 (모든기간) One-Hot 임베딩을 적용한 모든 데이터.

학습시간 : (AMD Ryzen 5 - 3600 , 3.6 GHz Cpu 6개 사용)

- 50회 트레이닝 : 3시간 30분

- 120회 트레이닝 : (대략) 9시간

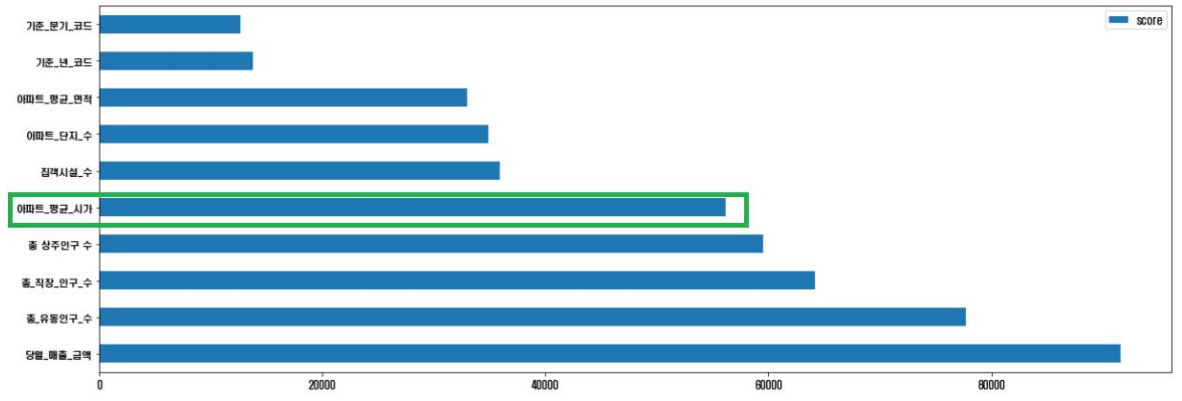


Xgboost 모델을 50회 학습하고 도출된 Feature 중요도.

(One-hot embedding을 적용한 상권코드의 경우 Weight가 0 으로써 그래프에서 제외하였음)

“아파트_단지_수”가 중요할것이란 초기 예상과 달리 “아파트_평균_시가”가 폐업률을 결정하는 5번째로 중요한 Feature로 결과가 나왔음.

금액 단위가 큰 아파트 “아파트_평균_시가” 특성상 모델 분석 결과에 영향을 끼친 것으로 판단하였음. 데이터 정규화(min-Max) 적용후, 다시 결과를 확인해봄.



데이터 정규화 적용 후, 50회 학습 Xgboost 모델 Feature 중요도 결과.

데이터 정규화 적용 유무와 관계없이, “아파트_평균_시가”는 폐업률을 결정하는 5번째로 중요한 Feature로 결과가 나옴.

```
In [47]:
1 from sklearn.metrics import mean_squared_error
2
3 np.sqrt(mean_squared_error(xgb_50_pred,norm_all_data[['폐업률']]))

Out[47]: 3.1385845340957617
```

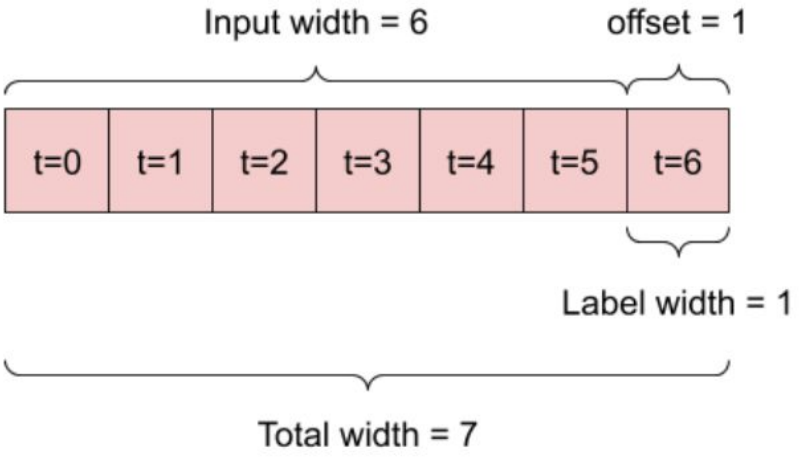
전체 기간, 모든 상권에 대해서 예측한 결과에 대한
RMSE 값 : 3.14

3. RNN(Recurrent Neural Network)을 사용한 시계열 예측

(1) 모델 선정 근거

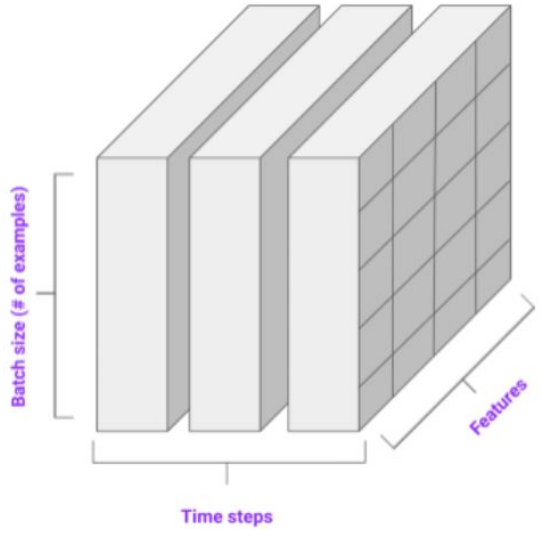
- RNN은 시계열을 단계적으로 처리하여 지금까지 학습한 정보를 요약하고 내부 상태를 유지하기 때문에, 시계열 데이터 분석에 적합한 모델이다.

- 시계열 데이터는 몇 개의 과거 데이터를 학습할 것인지, 그리고 얼마나 멀리 있는 예측을 배워야 하는지에 따라 time window로 분할할 수 있다. 예를 들어 지난 6개의 과거 데이터로부터 다음 1개의 값을 예측한다면 아래 그림과 같이 표현되며 이와 같은 구조를 갖도록 데이터를 시간정보를 바탕으로 샘플링하는 time window 함수를 정의하면 효과적으로 시계열 데이터를 다룰 수 있다.



(2) 모델 기능 정의

- 모델은 지난 1년 (4개 분기)의 데이터를 학습하여 다음 1개 분기의 폐업률을 예측한다고 가정한다. Time window로 샘플링 된 데이터가 LSTM이라는 RNN의 layer에 학습하게 되는데, 아래의 그림과 같이 time step 별로 추출된 샘플(batch size) 데이터들을 학습한다.

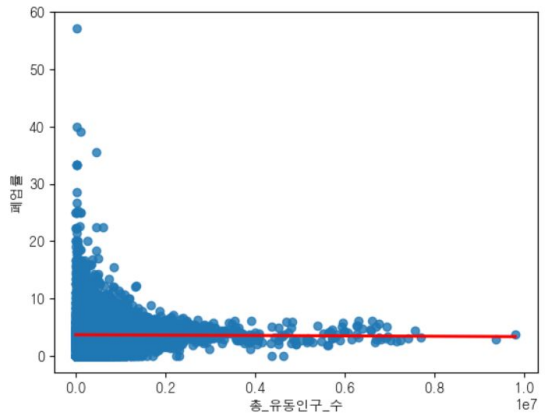
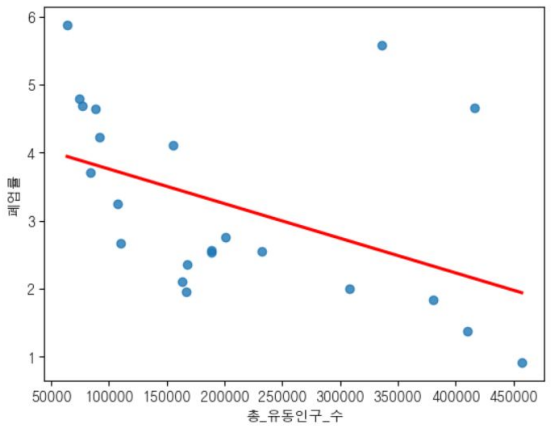


(3) 현재 진행 단계

- Time window를 반환하는 함수 정의, LSTM 모델의 파라미터(epoch 수, layer 수, activation function의 종류 등) 설정 중

4. 선형회귀 모델 (Linear Regression)

(1) 모델 선정 근거

	
<전체 상권>	*<특정 상권 내>*

- 위 그림처럼 전체 상권 관점에서 보면 7개의 상권 정보(총 유동인구 수, 집객시설 수 등)는 상권의 폐업률과 관계가 없어 보이나, 개별 상권 관점에서는 일부 특성들이 상권의 폐업률과 일정한 상관관계를 보이고 있음

- 산점도를 그려 시각화한 결과 일부는 선형적인 관계를 보이고 있음

- 따라서 선형회귀 모델을 통해 상권별 폐업률 예측 모델을 도출하고자 함

(2) 다중선형회귀 모델

- 선형회귀 모델 : 선형적 관계에 있는 독립변수와 종속변수에 대해, 독립변수가 변할 때 종속변수가 어떻게 반응하는지 살펴봄으로써 두 변수 사이의 관계를 하나의 식으로 표현하고자 하는 모델

- 회귀 모델의 목표는 종속변수를 가장 잘 설명할 수 있는 최선의 회귀 직선을 찾고, 이를 바탕으로 새로운 값에 대응하는 예측을 하는 것이다.

- 선형회귀 모델 중, 여러 개의 독립변수를 바탕으로 회귀식을 도출하는 '다중선형회귀 모델'을 사용함

- 다중선형회귀에서의 추정 회귀 직선 : $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \ \ldots\ldots \ \hat{\beta}_n x_n$

- 회귀식 도출을 위해 '최소제곱법'을 사용

- 최소제곱법 : 잔차 $(\varepsilon = y - \hat{y})$ 의 합이 최소가 되도록 하는 회귀직선을 찾는 방법

(잔차 : 실제 종속변수의 값과 추정 회귀 직선에서의 종속변수 값의 차이)

(3) 모델링 과정

- 파이썬 Scikit-learn 라이브러리 활용

a. 데이터 준비

```
# 상권별로 구분
for i in range(1496):
    globals()['df_{}'.format(i)] = df[df['상권코드_'+str(i)] == 1]
```

```
df_0
```

	Unnamed: 0	기준_년_코드	기준_분기_코드	상권코드_0	상권코드_1	상권코드_2	상권코드_3	상권코드_4	상권코드_5	상권코드_6	...	상권코드_1495	총_유동인구_수	아파트_단지_수	아파트_평균_면적	아파트_평균_시가	총 상주인구 수	집객시설_수	당월_매출_금액	총_직장_인구_수	폐업률
1474	1474	2020.0	2.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	308310.0	26.0	69.0	249322039.0	1800.000000	73.0	4.046282e+09	842.0	2.000000
2949	2949	2020.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	336343.0	26.0	69.0	249322039.0	1800.000000	73.0	5.560254e+09	842.0	5.583756
4424	4424	2019.0	4.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	457213.0	29.0	75.0	249484517.0	1800.000000	73.0	6.001523e+09	842.0	0.913242
5917	5917	2019.0	3.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	409966.0	29.0	75.0	249484517.0	1806.000000	73.0	3.751288e+09	609.0	1.382488

- 성능 확인을 위해 149개 상권 중 하나의 상권을 임의로 선정하여 진행

b. 데이터 전처리

- 현재 데이터의 각 항목들은 금액, 사람수 등 다양한 단위로 구성됨

- 또한, 단위가 금액인 항목의 경우 절대적인 값의 크기가 다른 값들에 비해 크므로 선형회귀분석 시 회귀식이 편향될 수 있음 (회귀식이 절대적 크기가 큰 값에 지나치게 의존함)

- 따라서, 항목 간 비교와 선형회귀분석의 정확도를 위해 각 항목들의 단위와 범위를 일치시킬 필요가 있음

- Min-Max 정규화 진행

- Min-Max 정규화 : 값의 분포는 변하지 않고 값의 범위만 0 ~ 1 사이로 고정함으로써 단위나 크기로 인한 회귀식의 편향을 방지함

```
# min-max scaling 진행
df_lm_n = df_lm.copy()
for col in df_lm_n.columns:
    if col == '폐업률':
        pass
    else:
        df_lm_n[col] = minmax_scaling(df_lm[col], columns=[0])
```

```
df_lm_n.head()
```

	총_유동인구_수	아파트_단지_수	아파트_평균_면적	아파트_평균_시가	총 상주인구 수	집객시설_수	당월_매출_금액	총_직장_인구_수	폐업률
1474	0.621303	0.0	0.0	0.997718	0.267055	1.0	0.193064	0.919926	2.000000
2949	0.692598	0.0	0.0	0.997718	0.267055	1.0	0.600618	0.919926	5.583756
4424	1.000000	1.0	1.0	1.000000	0.267055	1.0	0.719405	0.919926	0.913242
5917	0.879839	1.0	1.0	1.000000	0.272903	1.0	0.113653	0.486034	1.382488
7410	0.897115	1.0	1.0	1.000000	0.272903	1.0	0.000000	0.486034	4.651163

c. 모델 생성

```
# 독립변수, 종속변수 설정
X = df_lm_n.drop('폐업률', axis = 1)
y = df_lm_n['폐업률']
```

```
# 학습 데이터, 검증 데이터 분할 (8:2)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 2)
```

```
# 모델 생성
model1 = LinearRegression()
```

```
# 모델 학습
model1 = model1.fit(X_train, y_train)
```

- 우선 폐업률을 종속 변수로 두고, 8개 상권 특성 모두를 독립변수로 둔 뒤 모델링 진행

```
# 회귀 계수
model1.coef_
```

```
array([  0.90733233,  27.46358137, -31.71061987,   5.42455323,
         0.73496505,  10.18879384,  -4.66487076,   2.54973784])
```

```
# 회귀 절편
model1.intercept_
```

```
-10.385670864892797
```

d. 모델 성능 확인

	y_test	y_predict
18299	4.102564	2.032825
1474	2.000000	7.420253
20985	3.703704	3.947820
10245	2.764977	-0.883457
27692	5.882353	4.132065

```
# 훈련 데이터 RMSE
y_pred = model1.predict(X_train)
math.sqrt(mean_squared_error(y_train, y_pred))
```

```
0.5521493426741524
```

```
# 검증 데이터 RMSE
y_pred = model1.predict(X_test)
math.sqrt(mean_squared_error(y_test, y_pred))
```

```
3.165347586107262
```

- 예측의 품질이 떨어짐

- 다중공선성 발생이 의심됨

(다중공선성 : 회귀분석에 사용된 모형의 일부 독립변수가 다른 독립변수와 상관 정도가 높아, 모델의 성능에 부정적인 영향을 미치는 현상)

- 훈련 데이터에 대한 RMSE와 검증 데이터에 대한 RMSE 차이가 크며, 훈련 데이터에 비해 검증 데이터의 RMSE가 훨씬 큼

- 과적합 문제가 의심됨

e. 모델 성능 개선

- 성능 개선을 위한 다양한 방법을 시도함

- 다중 공선성 제거를 위한 VIF(분산 팽창 요인) 계산 및 조치

(VIF : 다중공선성을 파악하기 위한 수치적 지표, 10 이상이면 위험, 5 이상이면 주의)

```
# VIF(분산 팽창 요인) 계산

from statsmodels.stats.outliers_influence import variance_inflation_factor

vif = pd.DataFrame()
vif["VIF Factor"] = [variance_inflation_factor(df_lm_n.values, i) for i in range(df_lm_n.shape[1])]
vif["features"] = df_lm_n.columns
```

```
vif
```

	VIF Factor	features
0	8.307699	총_유동인구_수
1	139.927796	아파트_단지_수
2	147.971756	아파트_평균_면적
3	30.885043	아파트_평균_시가
4	5.529633	총 상주인구 수
5	24.980176	집객시설_수
6	6.588330	당월_매출_금액
7	38.749139	총_직장_인구_수
8	12.034270	폐업률

- 변수 선택 알고리즘 적용 (RMSE 기준 최적 모형 선택법)

	model	rmse	features
1	LinearRegression(copy_X=True, fit_intercept=Tr...	1.051952	[아파트_평균_시가]
2	LinearRegression(copy_X=True, fit_intercept=Tr...	1.027124	[총_유동인구_수, 아파트_평균_시가]
3	LinearRegression(copy_X=True, fit_intercept=Tr...	1.031313	[총_유동인구_수, 아파트_평균_시가, 당월_매출_금액]
4	LinearRegression(copy_X=True, fit_intercept=Tr...	1.073429	[총_유동인구_수, 아파트_평균_시가, 집객시설_수, 당월_매출_금액]
5	LinearRegression(copy_X=True, fit_intercept=Tr...	1.174115	[총_유동인구_수, 아파트_평균_시가, 총 상주인구 수, 집객시설_수, 당월_매출_금액]
6	LinearRegression(copy_X=True, fit_intercept=Tr...	1.418366	[총_유동인구_수, 아파트_평균_시가, 총 상주인구 수, 집객시설_수, 당월_매출_...
7	LinearRegression(copy_X=True, fit_intercept=Tr...	2.009890	[총_유동인구_수, 아파트_단지_수, 아파트_평균_면적, 아파트_평균_시가, 총 상...
8	LinearRegression(copy_X=True, fit_intercept=Tr...	3.165348	[총_유동인구_수, 아파트_단지_수, 아파트_평균_면적, 아파트_평균_시가, 총 상...

f. 모델 선택

- 상권 0에 대한 폐업률 예측 모델로 총 유동인구 수, 아파트 평균 시가 항목을 독립변수로 사용한 모델을 선정

```
model5 = LinearRegression().fit(X = X_train_5, y = y_train_5)
```

```
model5.coef_
```

```
array([-0.74588971, -0.83152059])
```

```
model5.intercept_
```

```
3.9040985049899692
```

- 도출된 회귀식 : $\hat{y} = 3.90410 + \ -0.74589x_1 + \ -0.83152x_2$

(x_1 = 총 유동인구 수, x_2 = 아파트 평균 시가)

	y_test	y_predict
18299	4.102564	3.323030
1474	2.000000	2.611052
20985	3.703704	3.625274
10245	2.764977	2.812509
27692	5.882353	3.812220

	VIF Factor	features
0	5.898906	총_유동인구_수
1	7.463135	아파트_평균_시가
2	1.938778	폐업률

- 기존의 모델보다 향상된 정확도를 보여줌

- 다중 공선성 위험이 매우 낮음

```
# 훈련 데이터 RMSE

y_pred = model5.predict(X_train_5)
math.sqrt(mean_squared_error(y_train_5, y_pred))
```

```
1.240457390525773
```

```
# 검증 데이터 RMSE

y_pred = model5.predict(X_test_5)
math.sqrt(mean_squared_error(y_test_5, y_pred))
```

```
1.0271238343650384
```

- 기존의 모델보다 훈련 데이터에 대한 RMSE가 증가했으나 훈련 데이터와 검증 데이터 모두에서 비슷한 수준의 RMSE를 보여줌

- 과적합 문제가 일부 해결

g. 모델의 한계

- 위 과정은 1개의 상권에 대한 모델만을 우선적으로 개발한 것이다. 총 1496개의 상권에 대한 모델 개발에는 상당한 시간과 노력이 필요할 수 있다.

- 여전히 예측 정확도가 불안정하다.

- 선형회귀 모형은 시간 개념을 포함시키기 못한다. 즉, 연도별·분기별 변화에 따른 값의 변동을 설명하지 못한다.

- 자기회귀 모형 등 시계열 회귀 모델에 대한 추가 조사가 필요하다.

II. 이후 진행 계획

1. 모델링 과정 완료 및 시각화 과정 시작

(1) Xgboost 모델

- Feature 중요도를 시각화 할 수 있는 툴인 Eli5 또는 Lime 에 대해서 적용

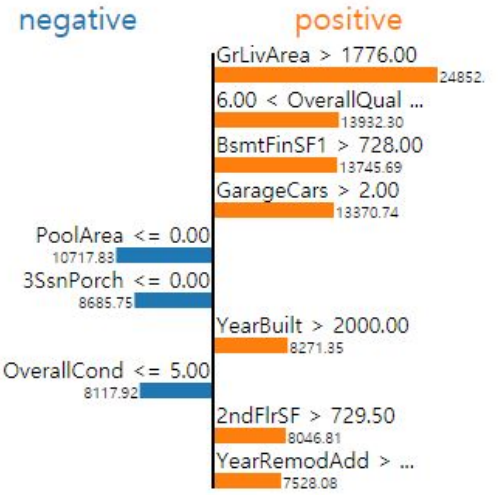
In [42]:
```
from eli5 import show_weights
show_weights(xgb_120)
```

Out[42]:

Weight	Feature
0.1051	당월_매출_금액
0.0156	상권코드_1433
0.0117	상권코드_1400
0.0115	상권코드_634
0.0105	상권코드_1271
0.0099	상권코드_804
0.0093	상권코드_1304
0.0092	상권코드_1424
0.0073	상권코드_523
0.0069	상권코드_452
0.0068	상권코드_312
0.0063	상권코드_1442
0.0061	상권코드_449
0.0059	상권코드_1059
0.0058	상권코드_13
0.0058	상권코드_81
0.0057	상권코드_75
0.0055	상권코드_739
0.0053	상권코드_818
0.0050	상권코드_1280

… 1486 more …

Eli5 예시

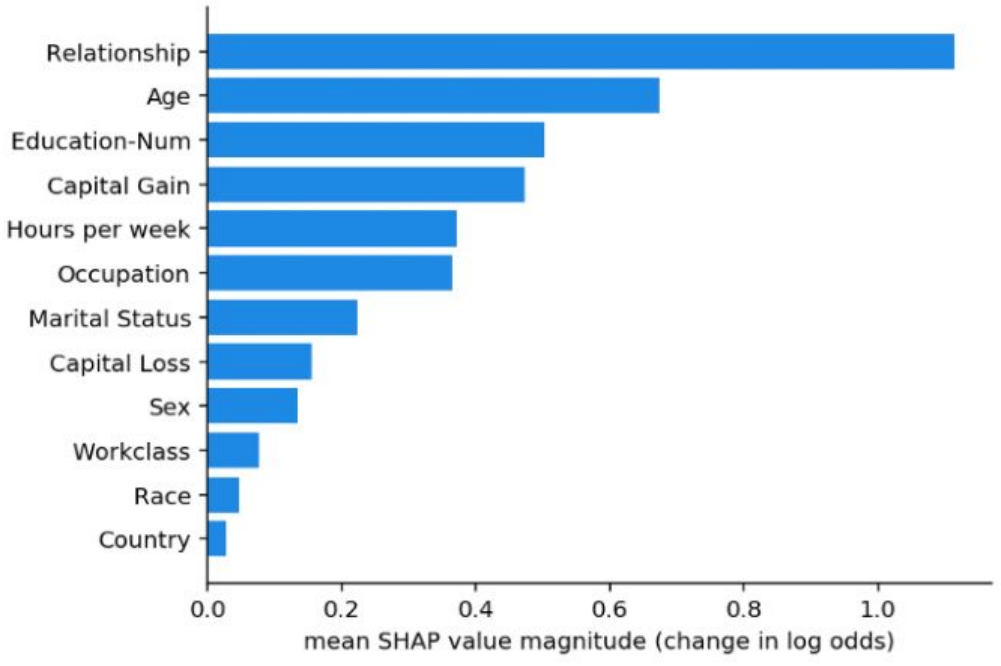


Lime 예시

- Xgboost 결과인 Feature 중요도의 신뢰성을 높이는 방법으로, Feature 중요도 결과에 SHAP value 를 적용하여 시각화

(Feature 중요도를 정하는 기준에는 일반적으로 쓰이는 “model(weight)”가 있지만 이 외에도 정보 획득량인 “gain” , “cover” 등 여러 가지가 존재한다.

SHAP value를 사용하는 것은 여러 기준을 종합적으로 판단한다는 의도가 있다.



Xgboost Feature 중요도에 SHAP value를 적용한 결과 예시

- 1496개의 상권별로 Xgboost 트레이닝 적용

- 과거 1년 or 2년 정보로 다음 분기 폐업률 예측하는 xgboost 모델 학습

(2) RNN 모델

- 모델의 폐업률 예측 성능을 MAE(Mean Absolute Error)로 계산, 과거 데이터로부터 예측한 폐업률 값 시각화

- LSTM layer의 가중치로부터 각 변수의 중요도를 비교 예정

- 상권 1496개 정보를 모두 제공하는 방법 고민

(3) 선형회귀 모델

- 모델의 예측 정확도를 높일 방법 고민 (추가적인 전처리)

- 시계열 개념이 포함된 자기회귀 모형 등에 대한 추가적 탐색 및 개발