청정한 제주도 만들기

01

어떻게 할까?

Q. 여러분의 평소 제주도 이미지는 어떤 가요?

형준 (팀장) lagojun1379@gmail.com

준수 iwicom@yonsei.ac.kr

유하린 dbgkfls99@naver.com

목차 INDEX

02

지금, 제주도의 상태는?

청정지역 제주도가 쓰레기 섬으로

03 문제정의

Q. 제주도의 현재 환경 상태와 그에 대한 원인 분석

한국의 보물, 제주도
많은 관광객들이 버리고 간 쓰레기로 인한 오염 심각.

청정 제주를 위한
2030 쓰레기 없는 제주 추진 선언.

1) 상관관계 분석을 위해서 날짜 관련 변수들 생성(연, 월, 일, 주, 주차, 시간 변화량)

```
alldata1['year'] = alldata1['base_date'].dt.year
alldata1['day'] = alldata1['base_date'].dt.day
alldata1['week'] = alldata1['base_date'].dt.week
alldata1['weekday'] = alldata1['base_date'].dt.dayofweek
alldata1['Week_num']=np.ceil(alldata1['day']/7)

alldata1['time'] = alldata1['base_date'].dt.date - alldata1['base_date'].dt.date.min()
alldata1['time'] = alldata1['time'].apply(lambda x : x.days)
alldata1 = alldata1.drop(columns=['base_date'])
```

2) 음식관련 카드소비에서 식품별 결제 건수와 결제 대금을 월별로 합산

month	emd_nm	use_cnt							...	use_amt						
		간식	농축수산물	마트/슈퍼마켓	배달	부페	식품	아시아음식	...	농축수산물	마트/슈퍼마켓	배달	부페	식품	아시아음식	양식
1	건입동	79.0	34.0	589.0	5.0	NaN	152.0	44.0	...	1824556.0	19891147.0	54010.0	NaN	6831676.0	1836340.0	2250744.0
1	구좌읍	79.0	25.0	329.0	5.0	NaN	231.0	25.0	...	1466175.0	11489793.0	66285.0	NaN	9145022.0	1627174.0	400165.0
1	남원읍	108.0	54.0	466.0	25.0	5.0	295.0	34.0	...	2199680.0	24828201.0	644192.0	294600.0	16392477.0	1854998.0	685927.0
1	노형동	1007.0	216.0	4905.0	123.0	15.0	943.0	363.0	...	12964266.0	168640726.0	2185932.0	2770222.0	32230959.0	16315635.0	9566325.0
1	대륜동	128.0	29.0	859.0	20.0	NaN	142.0	64.0	...	354993.0	37971731.0	396728.0	NaN	3178292.0	3405576.0	2081644.0

3) 내국인 유동인구에서 제주인, 타지인, 성별을 기준으로 거주인구, 방문인구, 근무인구 변수를 생성

	base_date	emd_nm	man_resd_pop_cnt	man_work_pop_cnt	man_visit_pop_cnt	woman_resd_pop_cnt	woman_work_pop_cnt	woman_visit_pop_cnt
0	2018-01-01	건입동	84744.8701	11638.6479	95737.5602	85324.4591	7335.6288	86268.1594
1	2018-01-01	구좌읍	142664.8188	9721.0341	116039.1218	126427.3259	7106.2017	126403.5696
2	2018-01-01	남원읍	152384.8823	10995.4261	108160.7162	132073.3097	6727.0772	105680.3226
3	2018-01-01	노형동	567232.7968	35126.5886	221249.7767	604264.9052	36706.3146	203360.6607
4	2018-01-01	대륜동	124968.7721	11301.7721	100900.0818	139747.2482	8660.3649	107820.8461
...	...	...	...	...	...	...	...	...
54906	2021-06-30	표선면	123817.9381	14018.5155	82469.6690	115466.4421	9502.4429	75581.9336
54907	2021-06-30	한경면	88602.5648	8791.6569	52753.3357	78815.2963	7237.9656	45812.9111
54908	2021-06-30	한림읍	247501.5803	31471.0957	148131.6335	222983.3985	17722.0108	138819.5661
54909	2021-06-30	화북동	277288.5428	30268.6911	138252.9350	295212.0508	19883.9228	102795.4549
54910	2021-06-30	효돈동	45364.8472	2687.9843	21888.4554	42956.9151	2603.2027	21640.7978

2021년 7월과 8월의 제주도 쓰레기 배출량 예측에 사용된 변수

	base_date	year	month	day	건입동	구좌읍	남원읍	노형동	대륜동	대정읍	대천동	도두동	동홍동	봉개동	삼도1동	삼도2동
0	2018-01-01	2018	1	1	1708250.0	NaN	1239600.0	9357900.0	1717700.0	1264950.0	1765600.0	613300.0	4165050.0	395300.0	2556750.0	1425050.0
1	2018-01-02	2018	1	2	1841150.0	NaN	1569850.0	10152950.0	2121750.0	1278200.0	2148050.0	748400.0	4361400.0	356300.0	2724050.0	1502850.0
2	2018-01-03	2018	1	3	1411450.0	NaN	1405650.0	8899800.0	1920300.0	1284450.0	2061650.0	538000.0	4028450.0	300400.0	2333350.0	1351550.0
3	2018-01-04	2018	1	4	1558700.0	NaN	1390250.0	9141400.0	1710000.0	1070000.0	1735000.0	444450.0	3725450.0	483350.0	2503300.0	1455150.0
4	2018-01-05	2018	1	5	1338350.0	NaN	1585700.0	8024400.0	1803600.0	1242000.0	1978850.0	333800.0	4205100.0	265300.0	2238050.0	1419650.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1272	2021-06-26	2021	6	26	1482150.0	1426300.0	1948850.0	7835650.0	1711650.0	3798350.0	1753950.0	685500.0	3434800.0	538300.0	2217450.0	800400.0
1273	2021-06-27	2021	6	27	1446550.0	1382150.0	2292900.0	9172600.0	1896400.0	3541700.0	2182150.0	987150.0	3748200.0	579250.0	2177350.0	831750.0
1274	2021-06-28	2021	6	28	1323550.0	1263300.0	2191650.0	8274150.0	1879450.0	3623300.0	2171950.0	604050.0	3595350.0	599500.0	1917150.0	770900.0
1275	2021-06-29	2021	6	29	1310650.0	1336600.0	2072350.0	8130450.0	1840100.0	3460250.0	1877550.0	655150.0	3526400.0	579750.0	1978050.0	810150.0
1276	2021-06-30	2021	6	30	1169850.0	1292600.0	2058200.0	7902000.0	1785400.0	3437350.0	1940950.0	597050.0	3244750.0	563300.0	1842400.0	818000.0

1277 rows × 46 columns

☑ 제주도 내 24개의 읍면동의 일별 쓰레기 배출량을 사용하였음.

예측에 사용된 모델

+

Prophet 라이브러리는 Facebook이 제공하는 라이브러리로, 비선형 데이터를 사용해 일간, 주간, 연간 주기를 기준으로 추세를 파악해 보여줌.

ARIMA모형은 ARMA모형이 과거의 데이터들을 사용하는 것에 반해 ARIMA 모형은 이것을 넘어서서 과거의 데이터가 지니고 있던 '추세(Momentum)'까지 반영한다.
이때, 우리는 ARIMA 모델에 계절적 성분이 추가 된 SARIMA 모델을 사용하였다.

→ 한가지 모델만 사용한다면 과적합 위험이 있기 때문에 두가지 모델을 사용하여 5대5 Blending하여 정답값 도출

```
In [14]: sub['year'] = sub['DateTime'].dt.year
         sub['month'] = sub['DateTime'].dt.month
```

```
In [15]: sub_prophet = sub.groupby(['year','month']).agg(sum).reset_index()
```

```
In [16]: sub2 = pd.DataFrame()
         sub2['DateTime'] = pd.date_range('2021-07-01','2021-08-31')
```

```
In [17]: df = train4[['Date','건입동']]
         df.columns = ['ds','y']
```

```
In [18]: model = Prophet()
         model.fit(df)
```

```
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
```

```
Out[18]: <fbprophet.forecaster.Prophet at 0x2a8cc88bd00>
```

1. dase_date를 기준으로 prophet 적용

```
In [19]: future = model.make_future_dataframe(periods = 62, freq = 'D')
         fcst = model.predict(future)
         pred_df = fcst.iloc[1277:]
```

```
In [20]: predict_ls = []
         for col in train4.columns[:-1]:
           fit = sm.tsa.statespace.SARIMAX(train4[col], order = (1,1,1), seasonal_order=(1,1,1,12)).fit()
           predict = fit.predict(start = 1276, end = 1276+61, dynamic = True).tolist()
           predict_ls.append(predict)
         #   print(f'{col} Forecasting....')

           temp = pd.DataFrame()
           temp['DateTime'] = pred_df['ds']
           temp.index = range(len(temp))

           temp[col] = predict
           temp2 = temp.groupby('DateTime')[col].sum().reset_index(name = col)

           sub2 = pd.merge(sub2, temp2, on='DateTime', how = 'left')
```

```
In [21]: sub2['year'] = sub2['DateTime'].dt.year
         sub2['month'] = sub2['DateTime'].dt.month
```

```
In [22]: sub_arima = sub2.groupby(['year','month']).agg(sum).reset_index()
```

```
In [23]: sub_pro_ari = sub_prophet.iloc[:,2:] *0.5 + sub_arima.iloc[:,2:]*0.5
```

```
In [26]: a = sub_pro_ari.apply(lambda x : round(x,1), axis = 1)
```

```
In [27]: a.to_csv('final_sub.csv', index = False)
```

2. Prophet 모델을 적용하기 위해 정해진 칼럼명으로 분석

```
In [11]: p = Prophet()
```

```
In [12]: sub = pd.DataFrame()
         sub['DateTime'] = pd.date_range('2021-07-01','2021-08-31')
```

```
In [13]: for col in train4.columns[:-1].tolist():
           train5 = train4[['Date', col]]
           train5.columns = ['ds','y']

           model = Prophet()
           model.fit(train5)

           future = model.make_future_dataframe(periods =365)
           fcst = model.predict(future)
         #   print(f'{col} Forecasting....')

           result = fcst.iloc[1276:]

           result.rename(columns={'ds':'DateTime'}, inplace = True)

           # display(result)
           temp = result.groupby('DateTime')['yhat'].sum().reset_index(name = col)

           sub = pd.merge(sub,temp, on = 'DateTime', how = 'left')
```

3. 일 별로 예측했기 때문에 다시 월 별로 합산

```
In [19]: future = model.make_future_dataframe(periods = 62, freq = 'D')
         fost = model.predict(future)
         pred_df = fost.iloc[1277:]
```

```
In [20]: predict_ls = []
         for col in train4.columns[:-1]:
           fit = sm.tsa.statespace.SARIMAX(train4[col], order = (1,1,1), seasonal_order=(1,1,1,12)).fit()
           predict = fit.predict(start = 1276, end = 1276+61, dynamic = True).tolist()
           predict_ls.append(predict)
         #   print(f'{col} Forecasting....')

           temp = pd.DataFrame()
           temp['DateTime'] = pred_df['ds']
           temp.index = range(len(temp))

           temp[col] = predict
           temp2 = temp.groupby('DateTime')[col].sum().reset_index(name = col)

           sub2 = pd.merge(sub2, temp2, on='DateTime', how = 'left')
```

```
In [21]: sub2['year'] = sub2['DateTime'].dt.year
         sub2['month'] = sub2['DateTime'].dt.month
```

```
In [22]: sub_arima = sub2.groupby(['year','month']).agg(sum).reset_index()
```

```
In [23]: sub_pro_ari = sub_prophet.iloc[:,2:] *0.5 + sub_arima.iloc[:,2:]*0.5
```

```
In [26]: a = sub_pro_ari.apply(lambda x : round(x,1), axis = 1)
```

```
In [27]: a.to_csv('final_sub.csv', index = False)
```

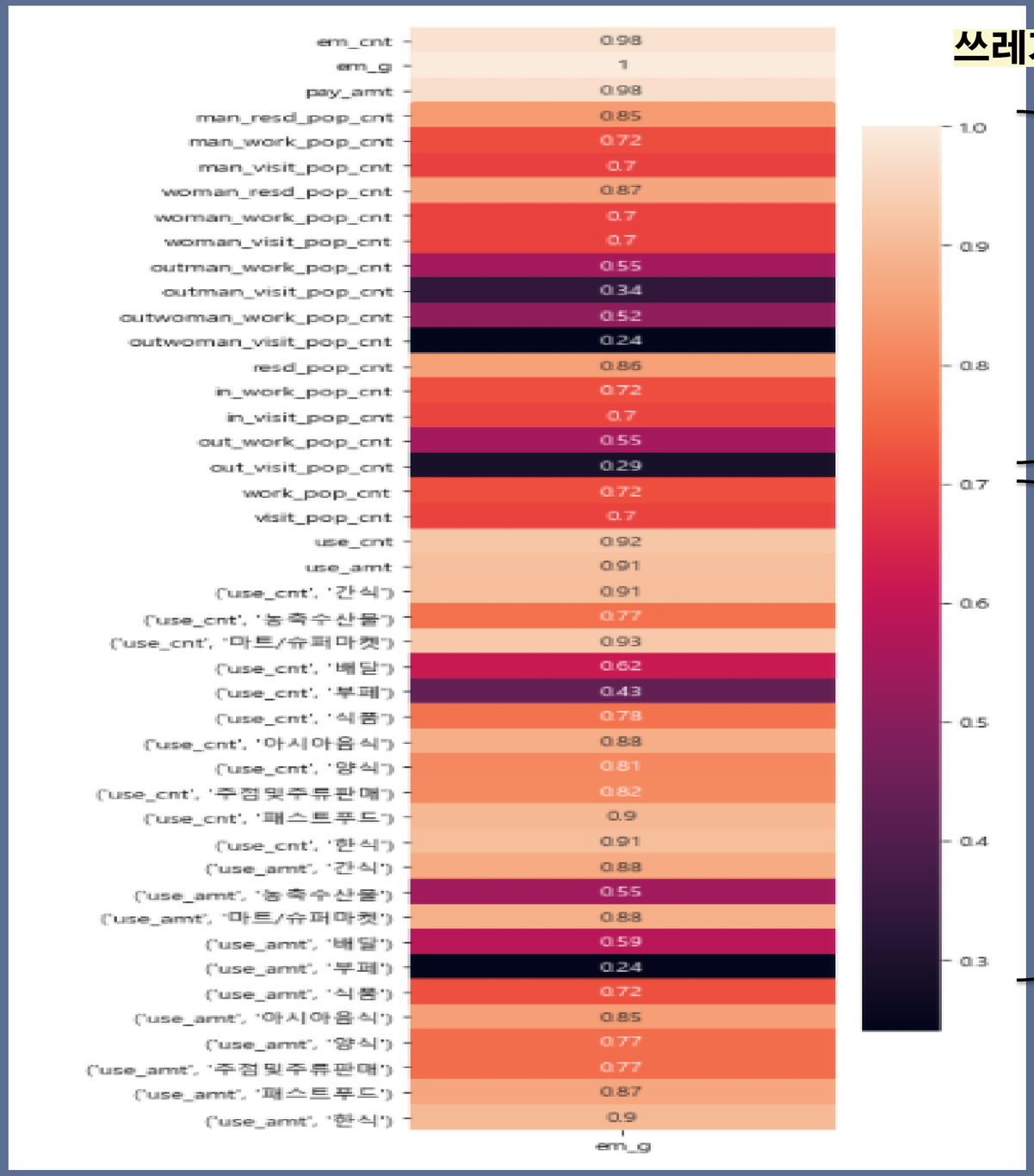
4. Sarima를 이용해서 예측,
   기존 arima에 계절성도 추가하였음, 일 별로
   예측했기 때문에 마지막에 월 별로 합산하였음.

5. 한 모델만 사용하면 과적합 위험성이 있어서
   prophet과 sarima 각자 예측한 값을 5:5
   blending해서 제출 파일을 만듬.

2021년 7월과 8월의 제주도 쓰레기 배출량 예측결과

지역	7월	8월
건입동	44159534.9	41417702.08
구좌읍	39534285.9	41668753.2
남원읍	72062677.24	73632759.19
노형동	279843947.9	276803588.3
대륜동	57040327.42	57065666.41
대정읍	107583851.8	112770696.7
대천동	67929047.15	70892567.06
도두동	22223296.43	22106911.38
동홍동	118059090.8	113970421
봉개동	17624702.27	16884385.71
삼도1동	66030391.72	64244651.42
삼도2동	32618921.44	29764633.88
삼양동	111007454.5	106477255.1
서홍동	47551993.39	52282061.98
성산읍	73867474.97	81140308.99
송산동	32364634.27	30566260.68
아라동	130576682.9	127363044.9
안덕면	46064938.62	48805201.18
알수없음	11659359.31	12064449.05
애월읍	102637585.7	103049399.1
연동	218257431.3	217977142.1
영천동	41337331.67	46512830.77
예래동	22699654.8	22928686.01
오라동	67934558.66	66541581.03
외도동	114757987.8	111723247.1
용담1동	35661596.5	35081182.16
용담2동	73822332.31	72722369.59
이도1동	38961806.9	36095201.21
이도2동	252130559.7	243354530.7
이호동	24774722.63	24570156.83
일도1동	16835165.73	15822134.85
일도2동	169150715.9	163318189.7
정방동	25948303.38	29111247.53
조천읍	64674084.24	66574927.31
중문동	75999822.13	77560341.33
중앙동	50173445.6	52209626.34
천지동	29885842.26	36518647.33
표선면	46446200.96	41583056.91
한경면	29992553.48	32131621.96
한림읍	87725270.63	91779002
화북동	124342588.8	118526798.7
효돈동	28459488.15	36429314.61

쓰레기 배출량과 상관관계 히트맵

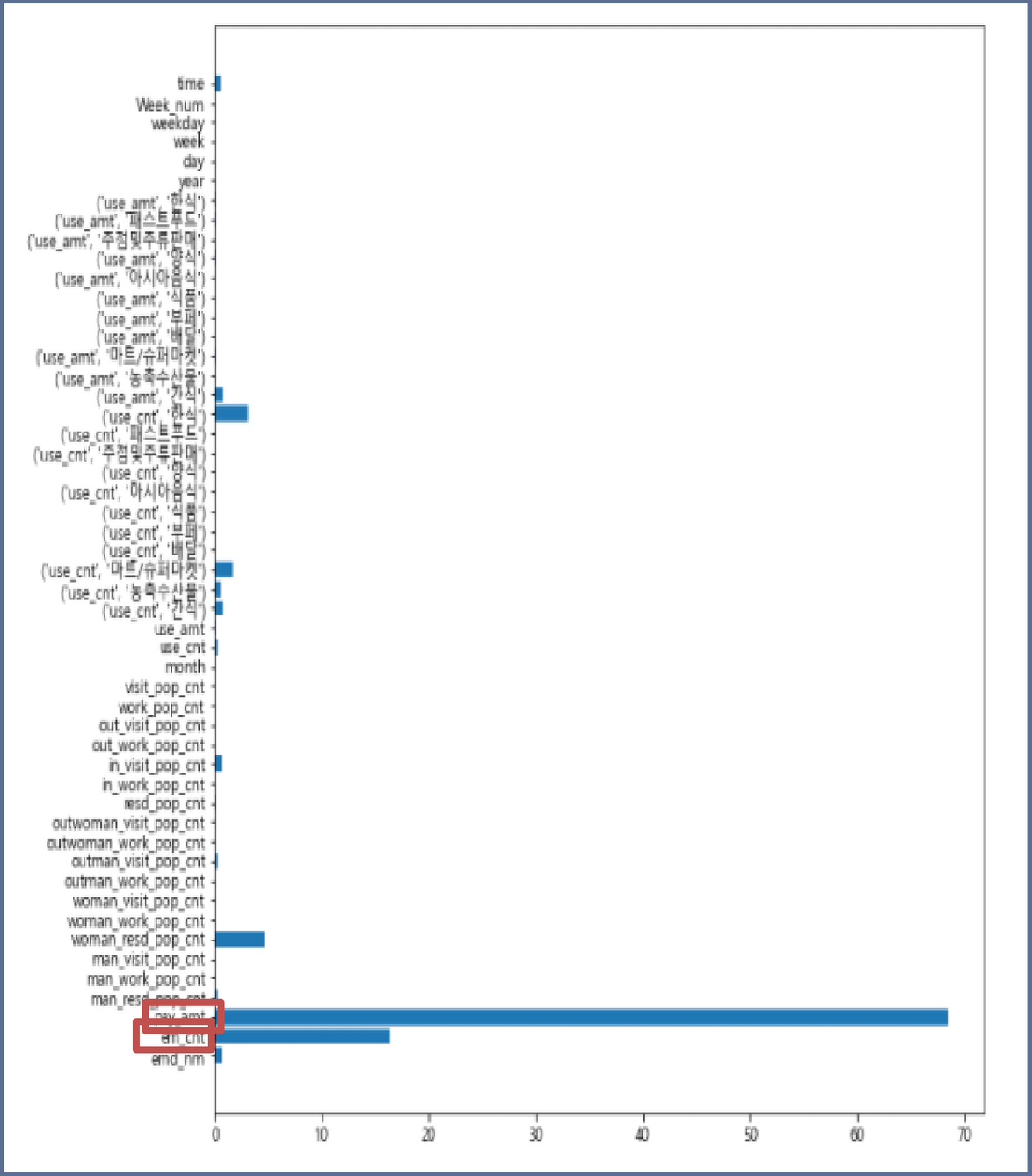


상관관계 0.7 이상 매우 강한 양의 상관 관계

상관관계 0.3이상 0.7이하 강한 양의 상관 관계

대부분의 변수들이 쓰레기 배출과 강한 양의 상관 관계가 있음을 확인

06 배출량 상관관계 확인 -변수의 중요도 확인



1. 범주형 변수들은 상관 관계를 볼 수 없음.

2. 전체 데이터를 hold-out 기법을 이용하여
3. Catboost model 적용을 한 후, 해당 모델 library의 feature importances 를 이용해서 모델이 판단한 변수 중요도를 확인하였음.

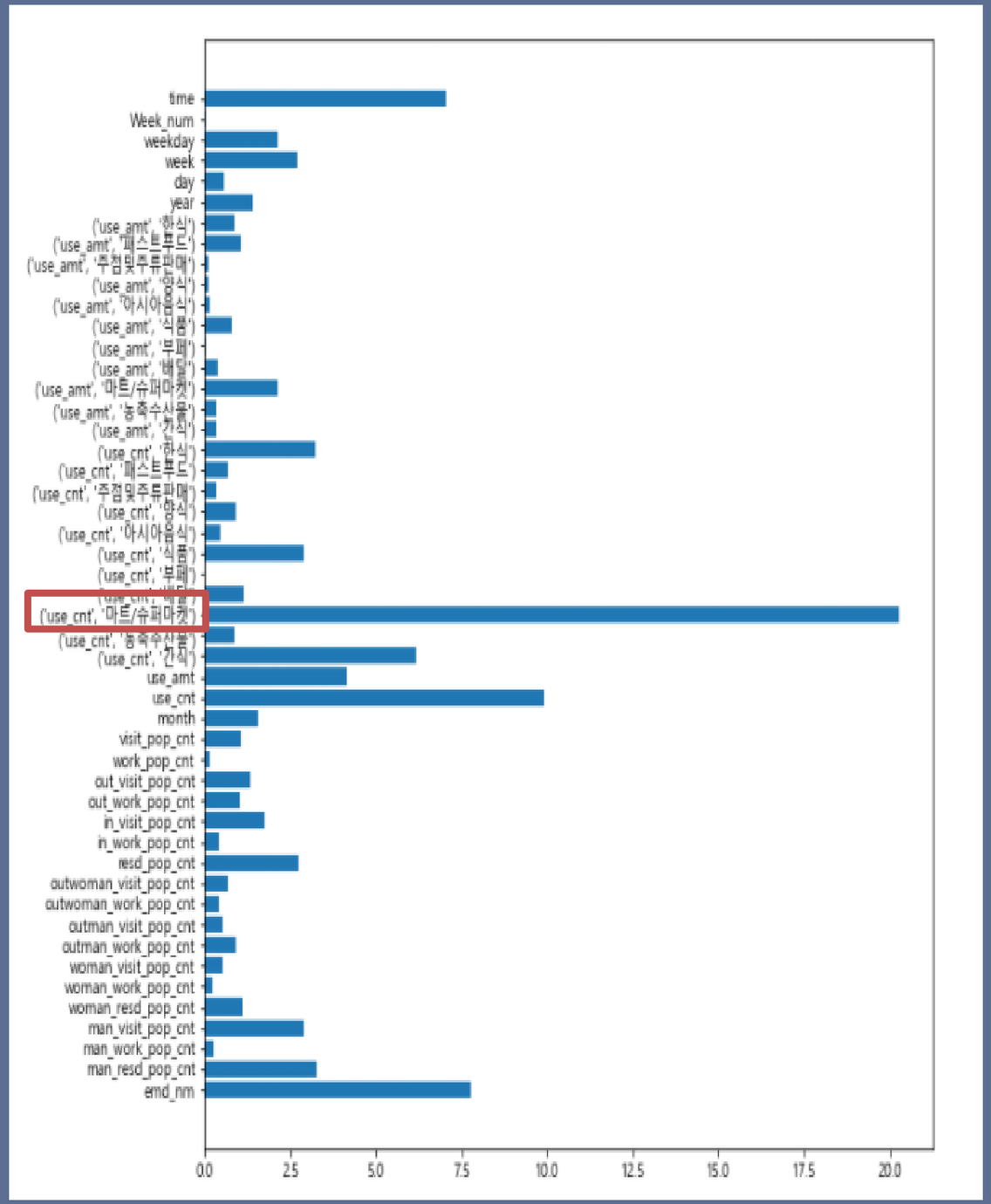
4. RMSE 는 cross-validation을 이용하여 구함.

```
rmse =  np.sqrt(-cvs)
rmse
```

```
97212.85322125202
```

5. RMSE 값은 좋은 값을 나타냈지만 두 변수 (pay_amt, em_cnt) 가 높은 연관성을 갖고 있으므로 쓰레기 배출량 증가의 원인을 분석하기에 적절하지 않다고 판단함.

06 배출량 상관관계 확인 -변수의 중요도 확인



1. 유효하지 않는 변수들을 제거하여 다시 부스트 모델의 feature_importances 를 사용하여 변수의 중요도를 파악.

```
rmse =  np.sqrt(-cvs)
rmse
```

```
481613.13185244054
```

2. RMSE 값은 전보다 높게 나왔지만 쓰레기 배출량과 관계가 높은 변수들을 찾아낼 수 있음.

제주도 쓰레기 배출량의 증가

그 원인은?

Q.쓰레기 배출량을 증가시키는 요인은 어떤게 있을까?

시간적요인

요일에 따라 쓰레기 배출량의 차이가 있다?

계절적요인

여름과 겨울의 쓰레기 배출량의 차이가 있다?

공간적 요인

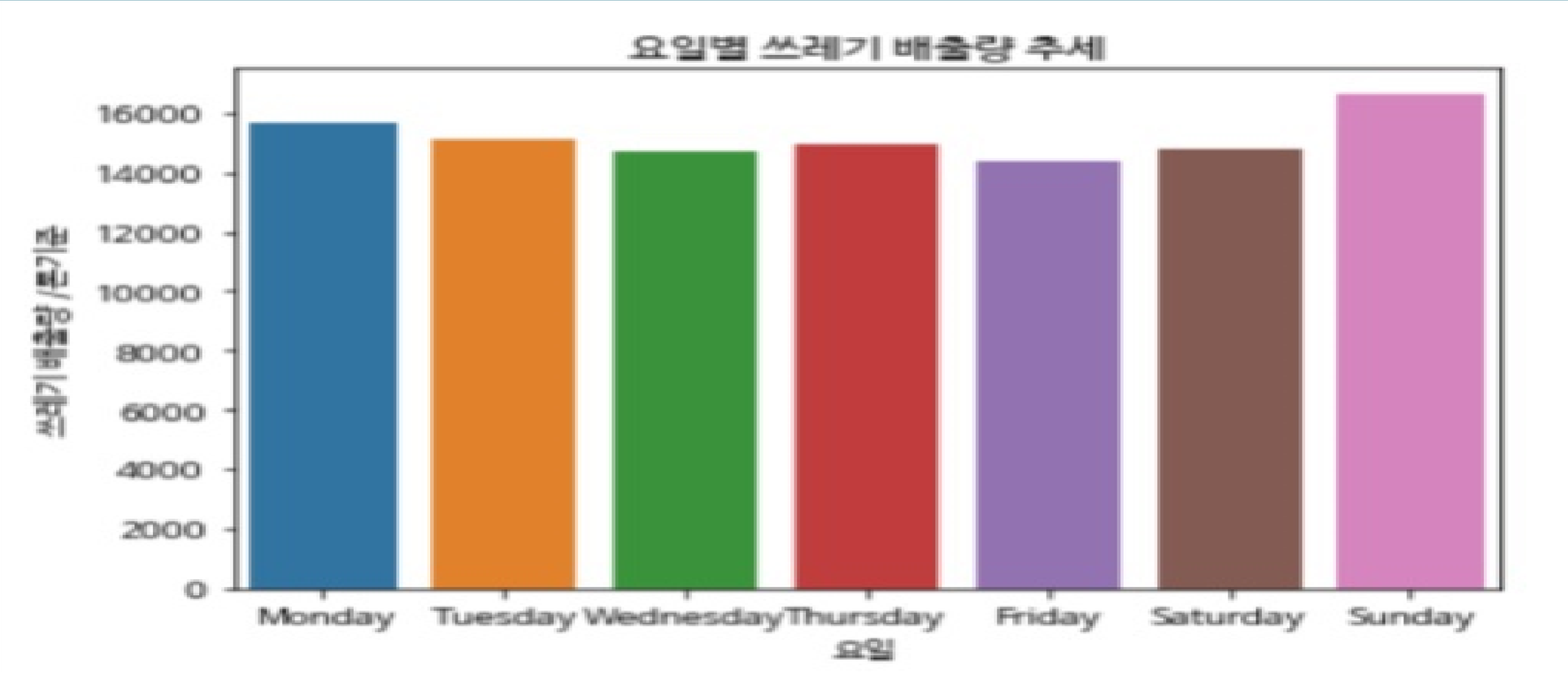
제주도 읍면동 별로 쓰레기 배출량의 큰 차이가 있다?

기타요인

쓰레기 배출량과 슈퍼마켓의 카드 소비량과 관계가 있다?

1)시간적 요인

요일별 배출량의 차이가 크지 않은 것으로 확인 되어짐. 즉, 요일별 변수는 배출량을 설명하기엔 부족함.

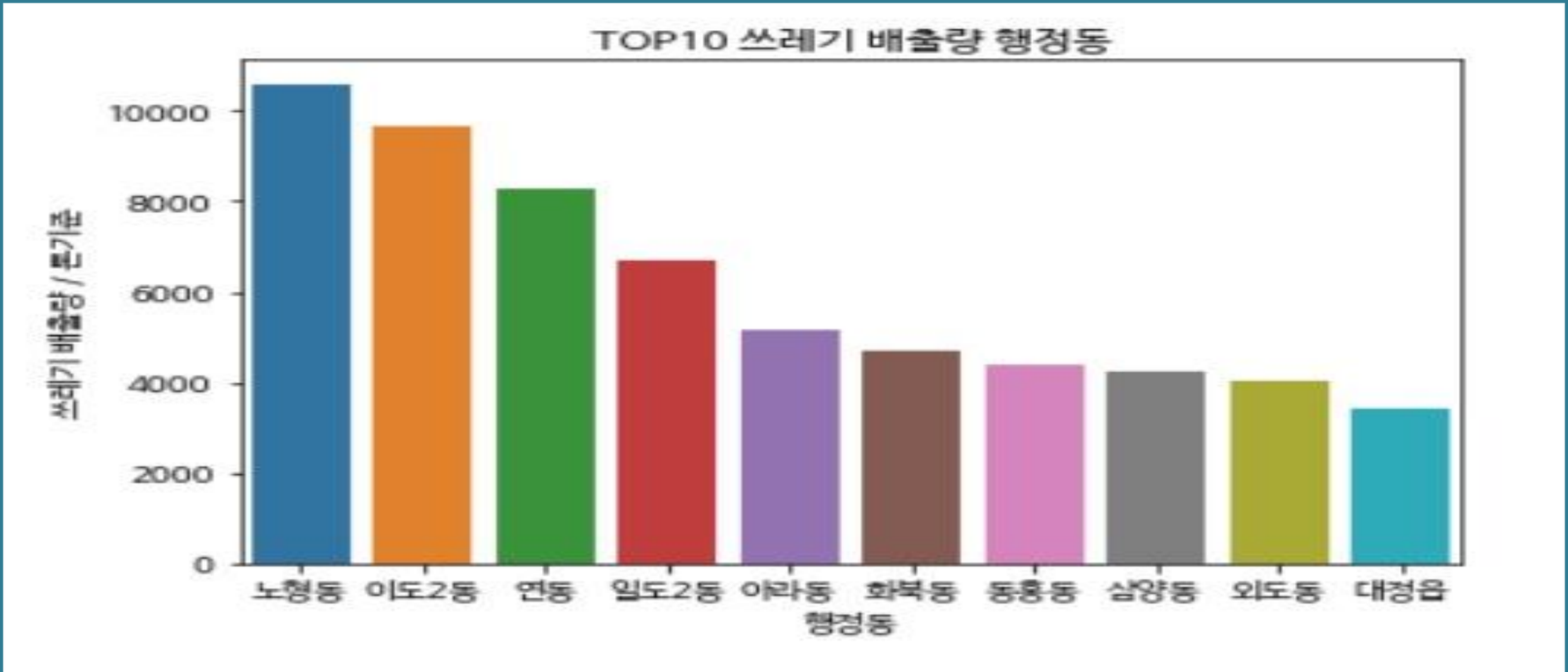


2)계절적 요인

2018~2021년 까지 3분기 및 여름이라는 계절적 시기에 쓰레기 배출량이 크게 증가하는 추세를 확인.

3)공간적 요인

행정동 별 기준에 따라 쓰레기 배출량이 높은 지역순으로 정리하여 배출량이 유독 높은 지역을 기준으로 쓰레기 배출량의 원인을 분석함.



2)공간적 요인-시각화

특정 시점인 2020년의 제주도 음식물 쓰레기 배출량을 제주도 지도에 맵핑하여 지역별로 쓰레기 배출량의 차이를 한눈에 볼 수 있도록 하였음.

해결책

사회에서 해결해요

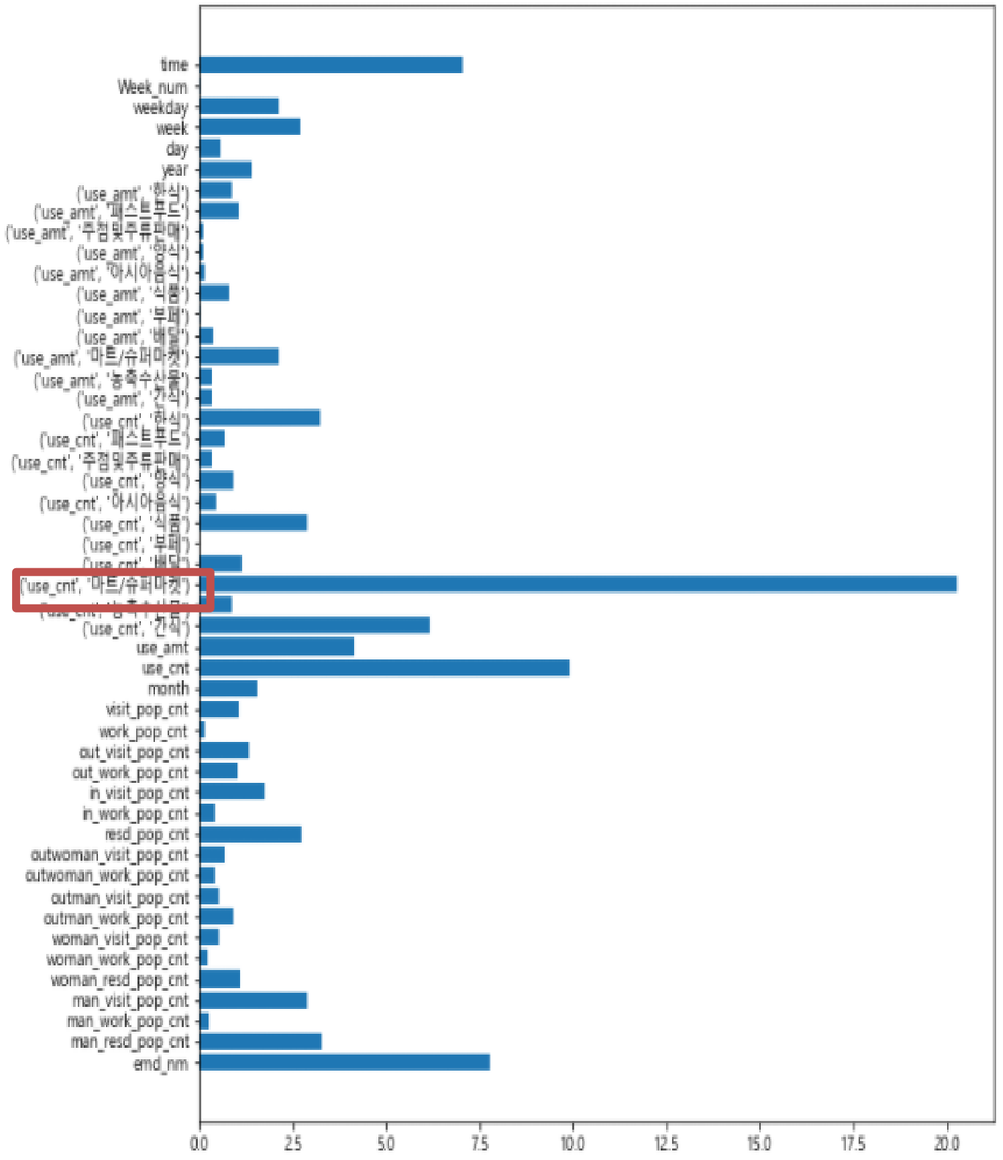
[홍보영상 만들기]
관광객들과 주민들에게 음식물쓰레기로 인한 제주도 피해의 심각성을 알리기

[인센티브 제공하기]
쓰레기 배출량이 많은 업종별로 월마다 쓰레기 배출량이 감소한다면 배달요금 감면 등의 인센티브를 제공하기.

일상에서 해결해요

[식판 사용하기]
식판 사용으로 먹을 만큼 덜어 먹어 음식물 쓰레게 줄이기

[음식물 쓰레기 분리수거 철저히 하기]
음식물 분리수거를 통한 온실가스와 매립량 감소시키기



쓰레기 배출량과 가장 연관이 높게 나온
슈퍼마켓의 카드 소비량을 이용하여
슈퍼마켓을 타겟으로 해결책을 제시함.

결론

1. 슈퍼마켓 곳곳에 제주도 음식물 쓰레기 증가로 인한 피해 상황을 표현한 포스터를 붙여 소비자들에게 경각심을 갖도록 한다.

2. 소비기한을 홍보하는 포스터를 슈퍼마켓의 음식 진열대 옆에 부착하여 폐기되는 음식물의 양을 줄인다.

감사합니다.