2022 FSI DATA CHALLENGE

-가계기반 소비 활동 지표를 통한 지역별 Segmentation 및 ESG 경영서비스 제안-

2022.8

접수번호 : T202206300062

목차

2022 FSI Data Challenge 분석 결과 보고서

1. 주제

주제	가계 기반 소비활동 지표를 통한 지역별 Segmentation 및 ESG 경영서비스 제안

2. 주제 선정 배경

본 대회를 준비하기 위해 금융에 관련된 다양한 기사와 자료들을 조사하던 중, 기존에 있는 지역별 경제 지표들은 고용, 물가, 생산 및 소비에 초점이 맞춰져 있다는 사실을 알게 되었다. 통계청에서 제공하는 '지역경제상황판'(이하 KOSIS)은 지역별 물가와 고용 등 경제지표를 인터넷에서 한눈에 볼 수 있는 서비스를 제공한다. KOSIS는 고용 부문에서는 경제활동참가율, 고용률, 실업률을, 물가 부문에서는 소비자물가지수, 생활물가지수, 신선식품물가지수를, 생산과 소비 부분에서는 광공업생산, 서비스업 생산, 소매판매액 지수를 제공한다. 이를 살펴본 결과 개인의 소비를 부분별로 나타내는 지표가 부족하다고 판단하였고, 대회 측에서 제공하는 라이프스타일 데이터를 토대로 가계 및 개인의 라이프 스타일 및 소비성향을 중심으로 하여 지역별 지표를 생성한다면 기존의 지표에서는 볼 수 없었던 새로운 내용을 표현할 수 있을 것이라 판단했다.

'2021 한국의 소비생활지표'(한국소비자원)에 따르면 국민이 체감할 수 있는 다양한 소비자 정책추진을 지원하고 국민 소비생활 향상에 활용될 수 있도록 한국의 '소비 생활 지표'를 격년을 주기로 발표해오고 있다. 이는 국민이 체감하는 소비생활 여건 인식, 만족도, 문제 경험 등을 측정, 분석한 것으로, 이 지표 데이터를 지역 소비생활 모리터링 및 지자체 과학적 정책 수립, 추진의 근거 자료나 교육, 정보 콘텐츠 및 정책 컨설팅 추진의 기초 자료로 이용하는 등 다양한 분야에 활용이 가

능하다는 사실을 알 수 있다. 이 보고서에 착안하여 우리는 우리가 만들고자 하는 가계 및 개인의 소비지표에 최근 몇 년간 기업들이 눈에 띄게 빠른 속도로 도입하고 있는 ESG 경영을 접목하여 새로운 서비스를 제안해 보고자 한다.

ESG란 환경보호(Environment), 사회 공헌(Social), 윤리 경영(Governance)의 약자로 지속가능한 경영의 비재무적 핵심 요소를 뜻한다. 이는 기업이 고객, 직원 등에게 얼마나 기여하는지, 환경에 대한 책임을 다하는지, 기업 운영과 지배구조가 투명한지 등을 다각적으로 평가하는 것으로 재무적인 이익만을 추구하는 것이 아니라 윤리적인 책임을 다하는 기업임을 표현할 수 있는 '사회적 책임투자'의 지표가 된다.

영국을 시작으로 스웨덴, 독일, 캐나다 등 이미 여러 나라에서 연기금을 중심으로 ESG 정보 고시 의무 제도를 도입했으며, UN은 2006년 출범한 유엔책임 투자원칙(UNPRI)을 통해 ESG 이슈를 고려한 사회 책임 투자를 장려하고 있다. 우리나라 또한 2025년부터 자산 총액 2조원 이상의 유가증권 시장 상장사의 ESG 공시 의무화가 도입되며, 2030년부터는 모든 코스피 상장사로 확대될 것이라고 발표하였다. 해외 및 금융권에서는 ESG를 친환경 기업의 대출금 증가 및 이자율 감소와 같은 사업 진행에 적용한다. 또한 개인에게는 그린카드와 같은 상품 출시를 통해 그린 소비를 장려하거나 자동차 배출량 감소 시 포인트를 적립해주는 등의 서비스를 제공한다. 따라서 본 대회의 데이터를 통해 기업 경영 트렌드인 ESG 경영을 우리가 만든 지역별 가계 소비 지표와 결부시켜 다양한 상품과 서비스를 제안한다.

3. 활용 데이터

본 분석에서는 신한은행 Grandata 3-1인 분기별 아파트별 라이프스타일 데이터를 메인으로 활용하여 개인의 소비지표를 선정하였다. 해당 데이터 이외에도 개인의 소비활동을 표현할 수 있는 변수를 더 많이 확보하기 위해 대회에서 제공하는 다양한 서브 데이터를 활용하였다. 활용 데이터는 다음과 같다.

데이터명	생성 및 차용 변수
Grandata 3-2	-
한국투자증권	국내주식, 해외주식, 채권, 연금
KB 손해보험	보험매출, 보험건수
KB카드 가맹점	교통/숙박/여행, 여가/오락, 의료, 의류/패션잡화

네 종류의 서브 데이터에서 개인의 소비활동 지수와 관련 있다고 판단되는 변수들을 추출했고 이를 다양한 방식으로 변형하여 사용했다. 한국투자증권 데이터에서는 ['국내주식'], ['해외주식'], ['채권'], ['연금'] 변수를 추출하였다. 이를 통해 얼마나 다양한 금융거래를 이용했는지 나타내는 ['금융거래다양성'], 금융거래를 얼마나 이용했는지를 전반적으로 나타내는 ['금융거래금액'] 변수로 변형해서 사용하였다. KB손해보험 데이터에서는 ['보험매출'], ['보험건수'] 변수를 추출해 사용하였다. KB국민카드 가맹점 데이터에서는 업종대분류를 시군구 기준으로 pivot_table하여 비슷한 업종별 매출금액을 나타내는 ['교통/숙박/여행'], ['여가/오락'], ['의료'], ['의류/패션잡화'] 변수를 생성하였다. Grandata 3-2에서는 따로 변수는 생성하지 않고 사용하였다.

앞의 데이터를 분석 목표에 맞추어 사용하기 위해 데이터 전처리 및 병합을 거쳐 마스터 데이터프레임을 생성하였다. 처리 과정 및 내용은 다음과 같다.

1) Python Groupby 함수를 통해 시군구를 기준으로 행들을 재조정한다. 이때 Aggregation Function은 Mean을 사용한다.
2) 데이터 병합을 위해 각 데이터의 중복 기간을 파악한다. 각 데이터의 수집 기간은 다음과 같다.

- ☐ Grandata 3-1: 2020년 1분기 ~ 4분기
- ☐ Grandata 3-2: 2020년 말 기준
- ☐ 한국투자증권: 2019년 1월 ~ 2021년 12월
- ☐ KB손해보험: 2020년 1월 ~ 2021년 4월
- ☐ KB국민카드 가맹점: 2019년 7월 ~ 2021년 6월

따라서 중복 기간인 2020년 10월, 11월, 12월을 기준으로 둔다.

3) 여러 데이터를 시군구 기준으로 병합하기 위해 띄어쓰기 수정, 시군구 범위 수정 등 시군구명을 메인 데이터인 Grandata와 통일하는 과정을 진행한다.

```
# 2020년 4분기 자료만 활용하기 위해 10 11 12 만 남김
kookmin_insurance = kookmin_insurance[kookmin_insurance['마감년월'] > 202009]


# 세종시 시군구에 대해 NaN값을 세종특별시로 변경
kookmin_insurance.loc[kookmin_insurance['광역시도'] == '세종', '시군구'] = '세종특별시'


# 결측치 처리
kookmin_insurance.dropna(subset=['광역시도', '시군구'] , axis = 0, inplace = True)

kookmin_insurance.loc[kookmin_insurance['시군구']== '수원시영통구', '시군구'] = '수원시 영통구'
kookmin_insurance.loc[kookmin_insurance['시군구']== '수원시팔달구', '시군구'] = '수원시 팔달구'
kookmin_insurance.loc[kookmin_insurance['시군구']== '고양시일산구', '시군구'] = '고양시 일산구'
kookmin_insurance.loc[kookmin_insurance['시군구']== '포항시남구', '시군구'] = '포항시 남구'
kookmin_insurance.loc[kookmin_insurance['시군구']== '부천시원미구', '시군구'] = '부천시 원미구'
kookmin_insurance.loc[kookmin_insurance['시군구']== '안양시동안구', '시군구'] = '안양시 동안구'

kookmin_insurance.loc[kookmin_insurance['시군구']== '부천시 소사구', '시군구'] = '부천시'
kookmin_insurance.loc[kookmin_insurance['시군구']== '부천시 오정구', '시군구'] = '부천시'
kookmin_insurance.loc[kookmin_insurance['시군구']== '부천시 원미구', '시군구'] = '부천시'

kookmin_insurance.loc[kookmin_insurance['시군구']== '전주시  완산구', '시군구'] = '전주시 완산구'
```

4) 수정된 시군구를 기준으로 중복 기간에 맞게 데이터를 병합한다.

5) 각 데이터의 기준인 시군구의 개수가 정확히 일치하지 않았기 때문에 발생한 결측치를 처리한다. 결측치 처리 과정은 다음과 같다.

1. MinMax_Scaler로 데이터 스케일링을 진행한다.
2. N_neighbor의 수를 default 값인 5로 지정한다.
3. KNN Imputation을 이용하여 결측값을 처리한다.
   - KNN Imputation은 분석 대상을 중심으로 가까운 K개 요소 중 가장 많은 수인 집단으로 분류하는 방법이다.
   - Mean, Max 등 Single Value Imputation보다 값을 다양하게 이용할 수 있고 Mode 알고리즘보다 정확하다는 장점이 있다.
4. Rescale을 통해 값을 복원한다.

4. 분석 결과

1) 지표 생성

주어진 데이터의 가계 소비를 기반으로 여러 분야의 새로운 소비지표를 생성하였다. 해당 지표는 각 분야의 소비성향 및 경제 활동성을 나타낸다.

가계 소비 지표	사용 데이터
쇼핑 소비지표	백화점 이용금액
	대형할인점 이용금액
	소형유통점 이용금액
	선호 쇼핑_대형할인점
	선호 쇼핑 백화점
	선호 쇼핑 브랜드
	선호 쇼핑_슈퍼마켓
	선호 쇼핑_아울렛
문화생활 및 자기 투자 소비지표	스포츠/문화/레저 이용금액
	미용 이용금액
	스타벅스 이용금액
	의류/잡화 이용금액
	유흥 이용금액
	게임 이용등급
	골프 이용등급
	선호 레저 이용수
	여가/오락
	의류/패션잡화
하이엔드 소비지표	하이엔드명품_정보
	하이엔드백화점_정보
	하이엔드_소비수준
	하이엔드_소득수준
	하이엔드_법인대표
	명품구매여부
여행 소비지표	숙박 이용금액
	여행 이용금액
	특급호텔 이용금액
	제주도지역 이용금액
	해외여행 이용금액
	국내여행 등급
	선호 관광지 사용자 수
	교통/숙박/여행
대중교통 소비지표	교통이용금액
	버스주중이용건수
	택시주중이용건수

	KTX주중이용건수
	버스주말이용건수
	택시주말이용건수
	KTX주말이용건수
	지하철 이용횟수
자차관련 소비지표	자동차판매이용금액
	자동차서비스/용품이용금액
	주유이용금액
	하이패스주중이용건수
	하이패스주말이용건수
	네비이용횟수
	자동차수입_정보
	차량 보유 수
	자동차이용여부
고객 신용 소비지표	주택담보대출잔액
	신용대출잔액
	분기신규주담대출여부
	분기신규신용대출여부
	최근차량할부약정액
재태크 관심 소비지표	주택보유건수
	증권사_정보
	암호화폐_정보
	금융거래 다양성
	금융 거래 금액
	국내주식
	해외주식
	채권
	연금
의료 관심 소비지표	의료이용금액
	보험 건수
	보험 매출
	의료
디지털 소비지표	배달앱 이용금액
	전자상거래 이용금액
	디지털 음악 이용
	앱 구매 등급

2) 각 지표의 특징

○ 쇼핑 소비지표

○ 문화생활 및 자기 투자 소비지표
-[선호 레저 이용수]: Grandata 3-2 데이터 기반으로 생성
-[여가/오락, 의류/패션잡화]: KB국민카드 가맹점 데이터 기반으로 생성

○ 하이엔드 소비지표
-명품 관련 소비/소득수준 표현

○ 여행 소비지표
-[선호 관광지 사용자 수]: Grandata 3-2 데이터 기반으로 생성
-[교통/숙박/여행]: KB 국민카드 가맹점 데이터 기반으로 생성

○ 대중교통 소비지표

○ 자차 관련 소비지표
-[차량 보유 수]: Grandata 3-2 데이터 기반으로 생성

○ 고객 신용 소비지표

○ 재테크 관심 소비지표
-한국투자증권 데이터 활용

○ 의료 관심 소비지표
-KB 손해보험 데이터와 KB 국민카드 데이터 활용

○ 디지털 소비지표

총 71개의 데이터 정보를 통해 10개의 가계 소비 지표를 생성했다.

3) 지수 생성

지역별 지수는 총 71개의 데이터 정보를 RFM Score 기법을 적용하여 생성하였다.

자세한 지수 생성 방법은 다음과 같다.

1. 지표 생성에 사용된 데이터 정보들을 MinMaxScale을 통해 0~1로 표현한다.

```
mm = MinMaxScaler()
data =  pd.DataFrame(mm.fit_transform(data),columns=data.columns)
data
```

SHC_SCRT1_SUM	SHC_SCRT2_SUM	SHC_SCRT3_SUM	SHC_SCRT4_SUM	SHC_SCRT5_SUM	SHC_SCRT6_SUM	SHC_SCRT7_SUM	SHC_SCRT8_SUM	...	금융 래
0.308240	0.343557	0.223126	0.507369	0.435160	0.339269	0.264049	0.246402	...	0.147
0.076073	0.086971	0.030040	0.072601	0.128056	0.084130	0.056546	0.049506	...	0.152
0.520256	0.372881	0.620622	0.374080	0.493902	0.784425	0.701569	0.787239	...	0.519
0.369035	0.252527	0.266922	0.391004	0.407327	0.450015	0.326764	0.305063	...	0.148
0.184772	0.380846	0.030930	0.216226	0.279885	0.231039	0.222227	0.159156	...	0.215
...	...	...	...	...	...	...	...	...	
0.230831	0.533367	0.048090	0.197227	0.394569	0.276472	0.221526	0.118296	...	0.109
0.452457	0.495263	0.371528	0.526443	0.702742	0.417485	0.413078	0.302699	...	0.121
0.114573	0.294047	0.056989	0.060420	0.240056	0.158313	0.113198	0.067371	...	0.058
0.131049	0.229828	0.011717	0.274393	0.135041	0.074779	0.072345	0.034980	...	0.077
0.114894	0.302910	0.064970	0.079384	0.182861	0.113362	0.081573	0.054765	...	0.075

2. 마스터 데이터프레임으로부터 각 지표를 만들기 위한 데이터 정보들을 추출한다.

```
shopping_df = data[['SHC_SCRT3_SUM','SHC_SCRT4_SUM','SHC_SCRT5_SUM','SP_SSM_CNT','SP_DPRT_CNT',
                    'SP_BRND_CNT','SP_SM_CNT','SP_OTLT_CNT']]
shopping_df.head()
```

시군구명	SHC_SCRT3_SUM	SHC_SCRT4_SUM	SHC_SCRT5_SUM	SP_SSM_CNT	SP_DPRT_CNT	SP_BRND_CNT	SP_SM_CNT	SP_OTLT_CNT
세종특별시	1.069760e+07	2.195265e+07	4.161617e+07	7.851613	1.290323	10.922581	164.767742	0.554839
가평군	1.440250e+06	3.141250e+06	1.358956e+07	2.812500	0.250000	6.875000	17.875000	0.125000
강남구	2.975520e+07	1.618556e+07	4.697701e+07	5.120567	6.049645	20.904255	143.042553	0.372340
강동구	1.279737e+07	1.691782e+07	3.907610e+07	10.973118	1.021505	13.483871	126.521505	0.526882
강릉시	1.482915e+06	9.355584e+06	2.744561e+07	4.591241	0.167883	17.868613	58.985401	0.116788

3. 추출된 데이터들을 모두 더하여 Margin이라는 새로운 열을 생성한다.

```
shopping_df['margin'] = shopping_df.sum(axis=1)
shopping_df
```

	SHC_SCRT3_SUM	SHC_SCRT4_SUM	SHC_SCRT5_SUM	SP_SSM_CNT	SP_DPRT_CNT	SP_BRND_CNT	SP_SM_CNT	SP_OTLT_CNT	margin
0	0.223126	0.507369	0.435160	0.226726	0.088660	0.194007	0.412697	0.057862	2.145607
1	0.030040	0.072601	0.128056	0.081215	0.017178	0.122114	0.040251	0.013036	0.504490
2	0.620622	0.374080	0.493902	0.147863	0.415681	0.371301	0.357613	0.038830	2.819892
3	0.266922	0.391004	0.407327	0.316863	0.070189	0.239500	0.315724	0.054946	2.062477
4	0.030930	0.216226	0.279885	0.132578	0.011536	0.317382	0.144486	0.012179	1.145203
...	...	...	...	...	...	...	...	...	...
220	0.048090	0.197227	0.394569	0.024939	0.006247	0.188923	0.205375	0.014221	1.079591
221	0.371528	0.526443	0.702742	0.356911	0.072834	0.308881	0.465103	0.246462	3.050904
222	0.056989	0.060420	0.240056	0.032208	0.034356	0.434486	0.125215	0.036099	1.019828
223	0.011717	0.274393	0.135041	0.040908	0.000000	0.361160	0.038244	0.026071	0.887535
224	0.064970	0.079384	0.182861	0.007219	0.021472	0.371892	0.092070	0.026071	0.845939

4. 비율로 표현하기 위해 각 데이터 정보들의 값을 Margin 값으로 나누어준다.

```
shopping_df.SHC_SCRT3_SUM = shopping_df.SHC_SCRT3_SUM/shopping_df.margin
shopping_df.SHC_SCRT4_SUM = shopping_df.SHC_SCRT4_SUM/shopping_df.margin
shopping_df.SHC_SCRT5_SUM = shopping_df.SHC_SCRT5_SUM/shopping_df.margin
shopping_df.SP_SSM_CNT = shopping_df.SP_SSM_CNT/shopping_df.margin
shopping_df.SP_DPRT_CNT = shopping_df.SP_DPRT_CNT/shopping_df.margin
shopping_df.SP_BRND_CNT = shopping_df.SP_BRND_CNT/shopping_df.margin
shopping_df.SP_SM_CNT = shopping_df.SP_SM_CNT/shopping_df.margin
shopping_df.SP_OTLT_CNT = shopping_df.SP_OTLT_CNT/shopping_df.margin
```

	SHC_SCRT3_SUM	SHC_SCRT4_SUM	SHC_SCRT5_SUM	SP_SSM_CNT	SP_DPRT_CNT	SP_BRND_CNT	SP_SM_CNT	SP_OTLT_CNT	margin
0	0.103992	0.236469	0.202814	0.105670	0.041322	0.090420	0.192345	0.026968	2.145607
1	0.059546	0.143909	0.253833	0.160984	0.034050	0.242054	0.079786	0.025839	0.504490
2	0.220087	0.132658	0.175149	0.052436	0.147410	0.131672	0.126818	0.013770	2.819892
3	0.129418	0.189580	0.197494	0.153632	0.034032	0.116123	0.153080	0.026641	2.062477
4	0.027008	0.188810	0.244398	0.115768	0.010073	0.277141	0.126187	0.010635	1.145203
...	...	...	...	...	...	...	...	...	...
220	0.044545	0.182687	0.365480	0.023100	0.005786	0.174995	0.190234	0.013172	1.079591
221	0.121778	0.172553	0.230339	0.116985	0.023873	0.101242	0.152448	0.080783	3.050904
222	0.055881	0.059245	0.235389	0.031582	0.033688	0.426038	0.122780	0.035397	1.019828
223	0.013201	0.309163	0.152153	0.046092	0.000000	0.406925	0.043090	0.029375	0.887535
224	0.076802	0.093841	0.216163	0.008534	0.025383	0.439620	0.108838	0.030819	0.845939

5. Entropy Weight를 이용하여 비율로 표현된 데이터에 곱할 가중치를 구한다.

```
shopping_weight
```

```
[0.09029234154905078,
 0.04016181455528776,
 0.021992788575943327,
 0.1316346151876212,
 0.2773473114067534,
 0.1171578185212953,
 0.07808559439542695,
 0.2433277158086213]
```

5-1) Entropy Weight 방식이란 새넌의 정보 이론을 바탕으로 지표의 속성정보를 활용하여 가중치를 산정하는 방법이다.

- Entropy란 정보 속성의 다양성으로 결정되며 지표 값의 응집도가 클수록 Entropy Weight가 높게 산정된다.
- 수학적으로 가중치를 산정하기 때문에 주관을 배제하고 객관적으로 가중치를 산정할 수 있다는 장점이 있다.

5-2) Entropy Weight 산정 과정은 다음과 같다.

5-2-1) Eq. (1)과 같이 자료를 행렬(D)로 구성한다. 이때 n은 세부지표 개수, m은 분석하고자 하는 지역 개수를 의미한다.

$$D = \begin{pmatrix} x_{1\cdot1} & \cdots & x_{1\cdot n} \\ \vdots & & \vdots \\ x_{m\cdot1} & \cdots & x_{m\cdot n} \end{pmatrix} \qquad (1)$$

5-2-2) 구축한 자료의 단위가 모두 다르기 때문에 자료를 정규화하기 위해 MinMax 방법을 사용한다. Entropy 이론에서 p는 발생 확률을 의미하지만 본 과정에서는 지표 값들을 정규화한 값을 나타낸다.

$$p_{i,j} = \frac{x_{i,j} - \min(x)}{\max(x) - \min(x)} \quad (2)$$

5-2-3) 셋째. 데이터를 정규화한 후, Eq. (3)과 같이 세부지표별 Entropy 값 (E_j)을 산정한다. Entropy값은 지표 값의 분산이 클수록 크게 산정된다. 이때 k는 대상지역의 개수를 고려하기 위한 상수이다

$$E_j = -k\sum_{i=1}^{m} p_{i,j} \ln p_{i,j} \quad (k = 1/\ln(m)) \quad (3)$$

5-2-4) 산정된 Entropy값을 활용하여 Eq. (4)와 같이 지표 속성 값의 다양성(d_j)을 산정한 후, Eq. (5)와 같이 지표별 가중치(w_j)를 산정한다.

$$d_j = 1 - E_j \quad (4)$$

$$w_j = \frac{d_j}{\sum_{j=1}^{n} d_j} \quad (5)$$

위의 과정을 함수로 생성하여 Entropy Weight를 산정하였다.

6. Pandas의 qcut을 이용하여 각 데이터 정보들을 다섯 등급으로 나누어준다.

```
shopping_df['department_store'] = pd.qcut(shopping_df['SHC_SCRT3_SUM'],q=5,labels=range(1,6)).astype(int)
shopping_df['large_discount_store'] = pd.qcut(shopping_df['SHC_SCRT4_SUM'],q=5,labels=range(1,6)).astype(int)
shopping_df['small_distribution_store'] = pd.qcut(shopping_df['SHC_SCRT5_SUM'],q=5,labels=range(1,6)).astype(int)
shopping_df['SP_SSM_CNT'] = pd.qcut(shopping_df['SP_SSM_CNT'],q=5,labels=range(1,6)).astype(int)
shopping_df['SP_DPRT_CNT'] = pd.qcut(shopping_df['SP_DPRT_CNT'],q=5,labels=range(1,6)).astype(int)
shopping_df['SP_BRND_CNT'] = pd.qcut(shopping_df['SP_BRND_CNT'],q=5,labels=range(1,6)).astype(int)
shopping_df['SP_SM_CNT'] = pd.qcut(shopping_df['SP_SM_CNT'],q=5,labels=range(1,6)).astype(int)
shopping_df['SP_OTLT_CNT'] = pd.qcut(shopping_df['SP_OTLT_CNT'],q=5,labels=range(1,6)).astype(int)
```

시군구명	SP_SSM_CNT	SP_DPRT_CNT	SP_BRND_CNT	SP_SM_CNT	SP_OTLT_CNT	department_store	large_discount_store	small_distribution_store
세종특별시	4	4	2	5	3	4	5	4
가평군	2	2	1	2	1	1	2	2
강남구	3	5	4	5	3	5	4	4
강동구	4	4	2	4	3	5	4	4
강릉시	3	2	3	2	1	2	3	3
...	...	...	...	...	...	...	...	...
홍천군	1	1	2	3	1	2	3	4
화성시	5	4	3	5	5	5	5	5
화순군	2	3	4	2	3	2	2	2
화천군	2	1	4	2	2	1	3	2
횡성군	1	2	4	2	2	2	2	2

7. 5에서 구한 Entropy Weight와 6에서 구한 등급을 곱하여 데이터별 지수를 구한다.

```
shopping_df['shopping_score'] = (shopping_weight[0]*shopping_df['department_store'] +
                                 shopping_weight[1]*shopping_df['large_discount_store']+
                                 shopping_weight[2]*shopping_df['small_distribution_store']+
                                 shopping_weight[3]*shopping_df['SP_SSM_CNT']+
                                 shopping_weight[4]*shopping_df['SP_DPRT_CNT']+
                                 shopping_weight[5]*shopping_df['SP_BRND_CNT']+
                                 shopping_weight[6]*shopping_df['SP_SM_CNT']+
                                 shopping_weight[7]*shopping_df['SP_OTLT_CNT'])
shopping_df.iloc[:,3:]
```

시군구명	SP_SSM_CNT	SP_DPRT_CNT	SP_BRND_CNT	SP_SM_CNT	SP_OTLT_CNT	department_store	large_discount_store	small_distribution_store	shopping_score
세종특별시	4	4	2	5	3	4	5	4	3.878103
가평군	2	2	1	2	1	1	2	2	1.596529
강남구	3	5	4	5	3	5	4	4	4.126032
강동구	4	4	2	4	3	5	4	4	3.823120
강릉시	3	2	3	2	1	2	3	3	2.122952
...	...	...	...	...	...	...	...	...	...
홍천군	1	1	2	3	1	2	3	4	1.759707
화성시	5	4	3	5	5	5	5	5	4.676116
화순군	2	3	4	2	3	2	2	2	2.532881
화천군	2	1	4	2	2	1	3	2	1.844523

8. 각 지수를 표준화한 후 100을 곱해주어 데이터값을 0-100으로 변형한다.

```
scaler = MinMaxScaler()
shopping = shopping_df.copy()
shopping = scaler.fit_transform(shopping)*100
```

```
shopping = pd.DataFrame(shopping)
shopping.columns = shopping_df.columns
shopping
```

JM	SP_SSM_CNT	SP_DPRT_CNT	SP_BRND_CNT	SP_SM_CNT	SP_OTLT_CNT	department_store	large_discount_store	small_distribution_store	shopping_score
86	75.0	75.0	25.0	100.0	50.0	75.0	100.0	75.0	72.939439
34	25.0	25.0	0.0	25.0	0.0	0.0	25.0	25.0	15.117759
72	50.0	100.0	75.0	100.0	50.0	100.0	75.0	75.0	79.222656
91	75.0	75.0	25.0	75.0	50.0	100.0	75.0	75.0	71.545995
00	50.0	25.0	50.0	25.0	0.0	25.0	50.0	50.0	28.458836
...	...	...	...	...	...	...	...	...	...
19	0.0	0.0	25.0	50.0	0.0	25.0	50.0	75.0	19.506607
77	100.0	75.0	50.0	100.0	100.0	100.0	100.0	100.0	93.163380
36	25.0	50.0	75.0	25.0	50.0	25.0	25.0	25.0	38.847627
11	25.0	0.0	75.0	25.0	25.0	0.0	50.0	25.0	21.402644
63	0.0	25.0	75.0	25.0	25.0	25.0	25.0	25.0	24.942787

9. 위의 과정을 반복하여 총 10개의 각 분야별 소비지표들의 지수를 생성한다.

	시군구	shopping_score	culture_self_score	high_end_score	travel_score	public_transportation_score	car_score	customer_credit_score	digital_score	medical
0	세종특별자치시	71.988913	37.368794	89.214669	30.237374	58.005815	80.718855	90.311823	56.797149	
1	가평군	53.480927	32.311972	67.882473	72.330108	44.067366	77.146772	55.521519	90.961575	
2	강남구	57.144903	19.481012	97.784690	14.510204	83.747071	76.818559	65.095277	76.949407	
3	강동구	69.847210	37.107985	88.992507	9.708241	71.511594	68.368701	85.637851	77.150137	
4	강릉시	29.007512	78.286959	63.938666	58.271614	59.855123	63.971518	42.118914	75.190461	
...	...	...	...	...	...	...	...	...	...	
220	홍천군	14.266303	42.807121	72.011608	52.563640	52.138099	79.201550	1.232999	86.188562	
221	화성시	89.106759	17.592056	86.280017	10.736587	78.295757	78.173373	79.804193	70.877188	
222	화순군	51.260225	50.385524	41.952838	58.975878	63.020360	64.223122	25.290768	74.306193	
223	횡성군	30.326563	63.527105	69.899348	57.630568	61.116569	75.447739	27.955996	83.990892	
224	흥덕구	55.367026	61.166526	72.838619	65.683879	54.180013	77.841098	1.232999	89.032505	

4) 지표 설명력 검증

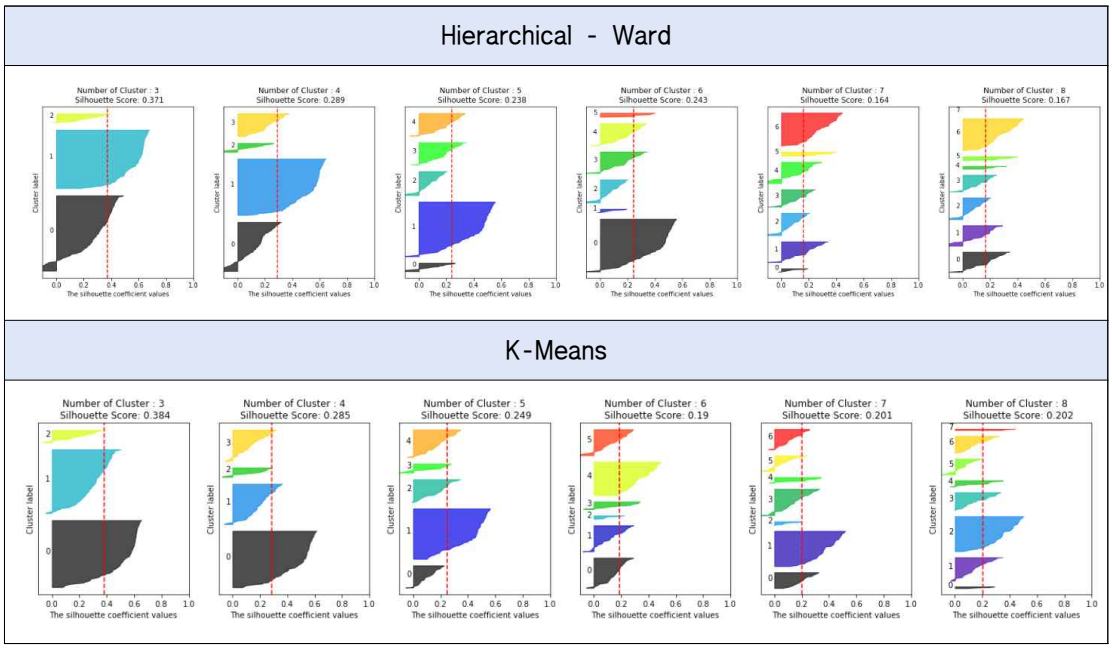
생성된 10개 지표의 설명력을 검증하기 위해 전체적인 소비활동을 표현할 수 있다고 판단한 Grandata 의 카드 소비금액을 임의의 y값으로 두었다. Stata의 OLS모델을 활용하여 선형회귀를 진행한 결과 R-squared 값은 0.828, Adj_R_Squred 값은 0.82로 본 분석을 위해 선정한 지표가 꽤 높은 수준의 설명력을 나타냄을 알 수 있다.

5) 클러스터링

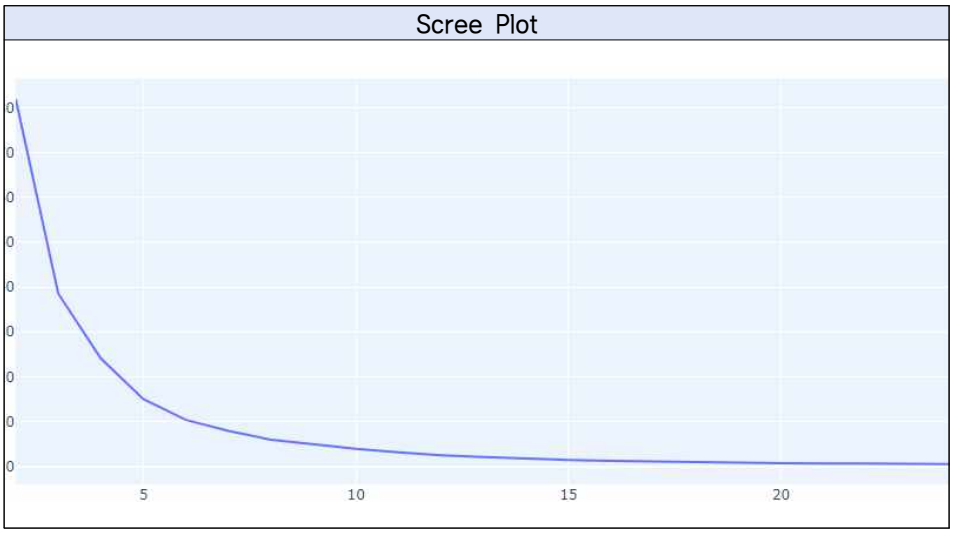
생성된 10개의 지표를 바탕으로 지역별 특징을 찾아 Segmentation을 진행하기 위해 머신러닝의 비지도 학습 중 하나인 Clustering을 사용하였다. Clustering 기법은 집단 간 정보와 분류 규칙 없이도 개체들이 다양한 특성 관계를 기반으로 이들을 유사 집단으로 분류할 수 있기 때문에 주어진 데이터 사이에서 의미 있는 자료 구조를 찾을 수 있다. 따라서 앞서 생성한 10개의 지표를 통해 지역들을 유사한 정도로 분류하고 해당 집단을 라벨링 함으로써 기존에 파악할 수 없었던 지역별 특징을 발견해냈다.

Clustering 방식에는 다양한 방법이 존재하는데, 많은 방법 중에서도 본 대회의 환경과 데이터의 양을 고려하여 Partitioning 방법 중 K-means Clustering, Hierarchical Clustering 중 ward Method, 밀도 기반 방법 중 DBSCAN 방법을 사용하였다. 세 가지 방법을 모두 진행한 후 군집화 평가지표 중 하나인 Silhounette Score와 형성된 군집의 개수를 참고하여 최적의 방법을 찾아내고 해당 방식으로 Clustering을 진행하였다.

앞서 위에서 언급한 지표를 구성하는 71개의 feature 들을 Clustering의 feature로 사용했다. 각 지역에 대해 최대한 많은 특징들을 반영하기 위해 지표 생성 후에 도출된 Score 값을 사용하지 않고 이와 같이 진행했다. 71개 Feature 들은 값의 Scale이 다르므로 거리를 기반으로 하는 Cluster 방법에 차질이 생길 수 있다. 따라서 모든 Feature 들에 대해 MinMax Scaling을 진행한 후 Clustering을 실행했다.



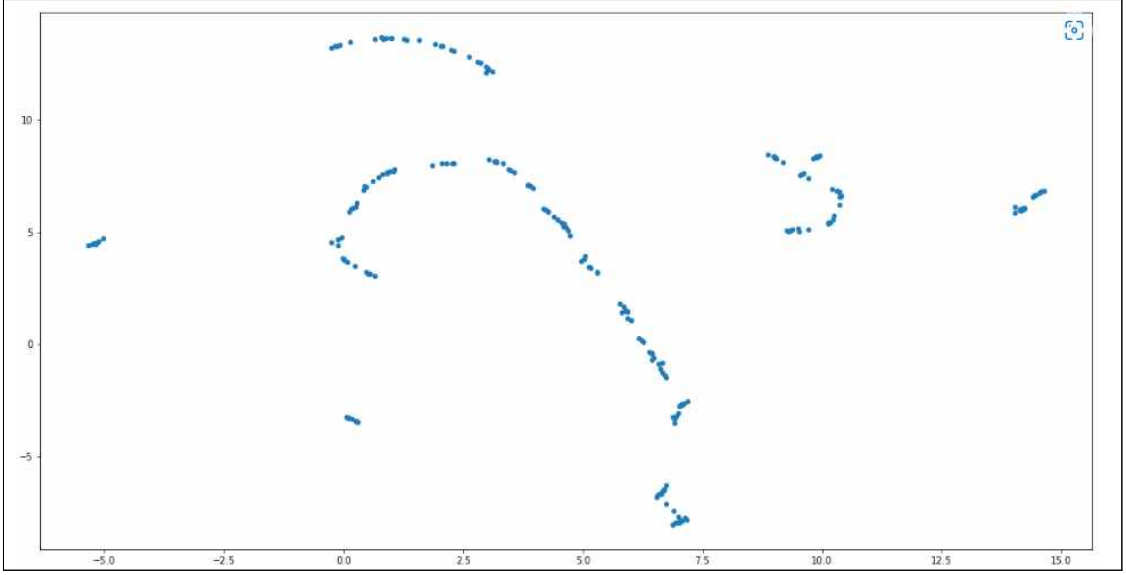
Silhouette Score와 형성된 군집의 개수를 확인한 결과 DBSCAN에서는 Silhouette Score가 0.4344로 가장 높게 나왔으나 군집의 개수가 2개로 형성되었기 때문에 거리 기반 방식을 사용하기로 했다. 거리 기반의 2가지 방법에서는 실루엣 계수가 비슷하게 산출되었다. 하지만 군집의 개수를 고려했을 때, Hierarchical 방법은 군집의 개수가 늘어남에 따라 Silhouette Score가 점점 내려가고 K-means 방법은 군집의 개수가 6개일 때 낮았다가 7개부터 다시 늘어나는 경향을 보였다. 따라서, Segmentation 해야 하는 지역의 개수가 225개인 것을 고려했을 때, 군집의 개수를 더 많이 사용할 수 있는 K-Means 방법을 사용하기로 했다.



K-Means를 사용하기 위해서는 선제적으로 K의 개수를 설정해야만 한다. 적절한 K 개수를 설정하기 위해 Scree Plot을 그려 K개 수가 변경될 때마다 비율 분산을 확인하여 기울기가 유의미하게 변하는 부분을 K로 설정하였다. Scree Plot의 결과와 앞서 진행한 Silhouette Score를 바탕으로 K(군집)의 개수를 7개로 설정하였다.

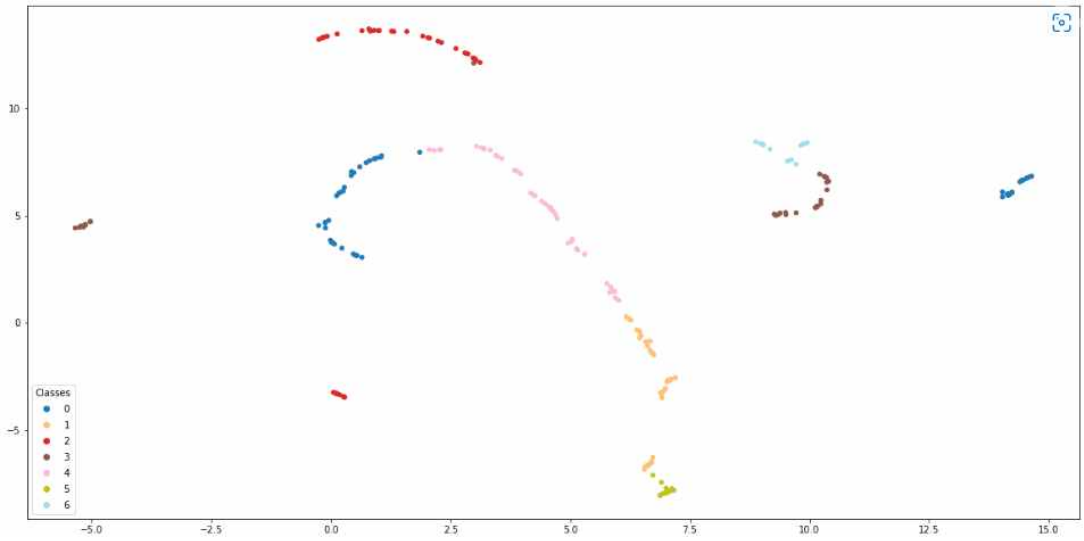
다음으로 K-Means 방법을 사용했을 때, 225개의 지역의 군집화가 잘 이루어졌는지 육안으로 확인하기 위해 차원 축소 방법의 하나인 UMAP을 활용해 시각화를 진행하였다.

UMAP 시각화를 통한 지역 밀집도 확인



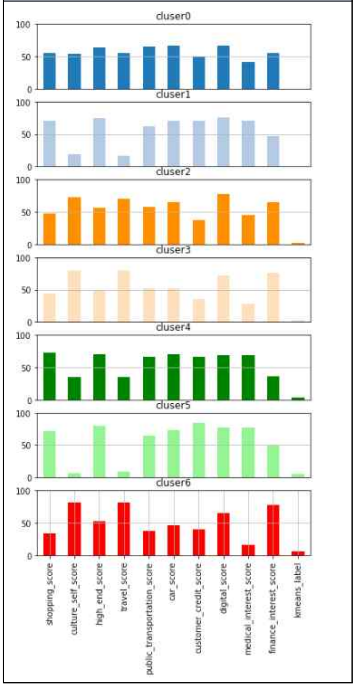
UMAP을 통해 71차원을 2차원으로 축소한 후 개체들의 밀집도를 확인한 결과 개체들이 어느 정도 군집을 이루고 있다는 것을 파악할 수 있었다. 여기에 K-Means를 통해 계산된 군집을 범주로 설정하여 다시 시각화를 진행하였다. 시각화 결과 밀집 정도에 따라서 군집들이 잘 형성되어 있는 것을 확인할 수 있었다.

UMAP 시각화를 통한 지역 밀집도 확인



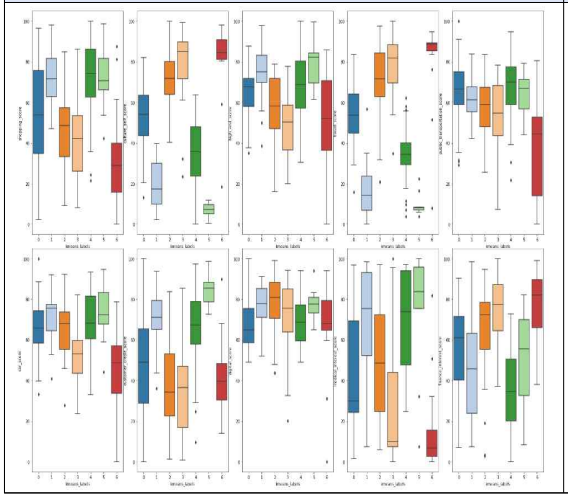
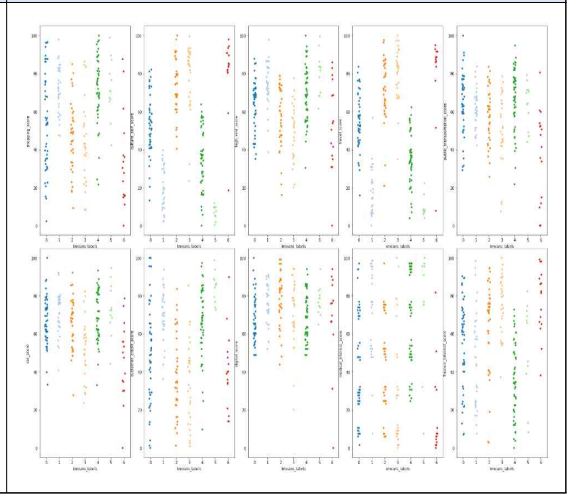
6) 군집 해석과 Segmentation

Cluster별 지수에 대한 평균 BarPlot 시각화

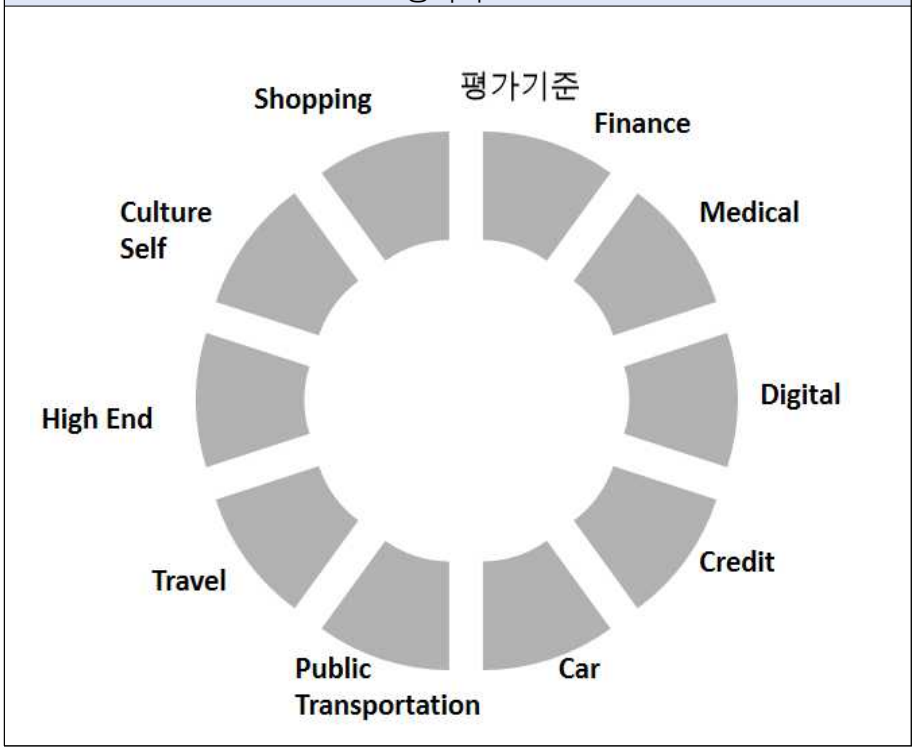


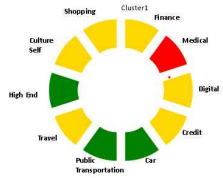
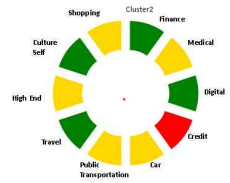
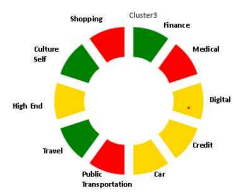
생성된 지표와 K-Means Clustering을 통해 만들어진 군집의 특성을 해석하고 지역별 Segmentation을 위해 시각화를 진행하였다. 각 Cluster 별로 지표가 상이하게 분포되어 있음을 확인하기 위해 Cluster 별 Bar Plot을 시각화하였다.

Cluster 별로 지표의 평균을 확인한 결과 1번, 4번 5번 클러스터의 지표 분포 양상이 비슷하고 2번, 3번 지표의 분포 양상이 비슷한 것으로 보인다. 0번 클러스터는 대부분 지표가 50 정도로 평균을 유지하고 있다. 6번 클러스터의 경우 다른 클러스터에 비해서 특징이 두드러지게 나타날 것으로 보인다. 더 자세히 특징을 파악하고 Segmentation 하기 위해 Box Plot과 Strip Plot을 통해 클러스터의 특징을 살펴보았다.

Cluster 별 지수 BoxPlot	Cluster 별 지수 StripPlot
	

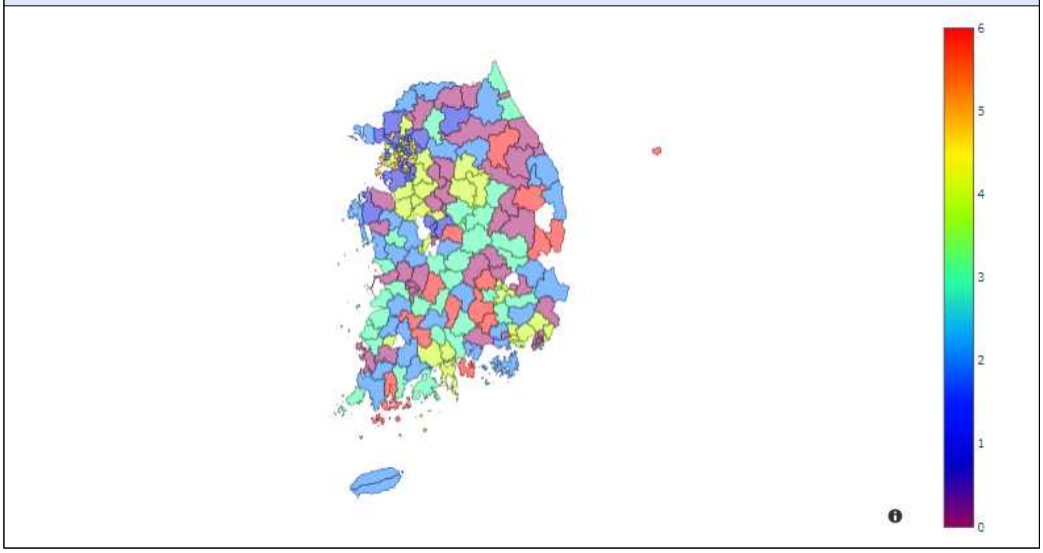
군집 별 지표별 box plot을 통해 중위수 및 1, 3 분위 수를 파악하고 Strip Plot을 통해 분산을 확인하여 군집 별 특성을 특정할 수 있었다. 특정 결과는 아래와 같이 표현한다. 각 Cluster 별 평가지표에 맞추어 해당 지표의 스코어가 높으면 초록색, 중간 정도면 노란색, 낮으면 빨간색으로 표시하여 설명에 쉽게 하였다.

평가지표


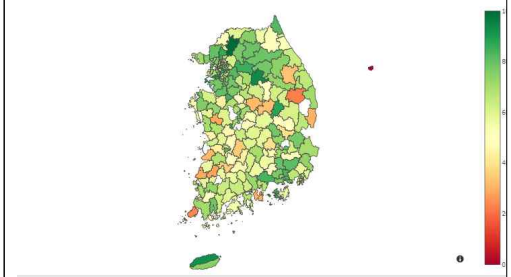
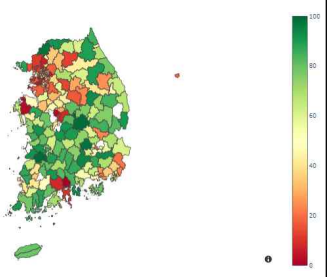
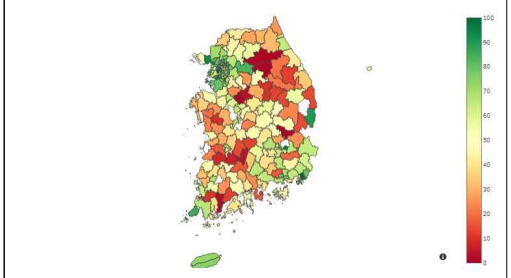
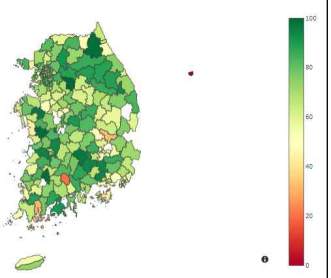
Cluster	특징
 0번	- 자차와 하이엔드, 대중교통에 대한 스코어 높음 - 의료에 대한 스코어낮음 - 자동차 및 이동수단에 특징 존재
 1번	- 쇼핑, 하이엔드, 자차, 의료에 대한 스코어 높음 - 재테크, 여행, 문화/자기소비에 대한 스코어 낮음
 2번	- 문화/자기소비, 여행, 재테크, 디지털에 대한 스코어 높음 - 신용 소비에 대한 스코어 낮음
 3번	- 문화/자기소비에 대한 관심 가장 높음, 여행, 재테크 스코어 높음 - 의료, 신용소비에 대한 스코어 낮음
 4번	- 쇼핑, 하이엔드, 대중교통, 자차, 신용소비에 대한 스코어 높음 - 여행, 문화/자기소비, 재테크에 대한 스코어 낮음 - 전반적인 스코어가 높은 경향
 5번	- 쇼핑, 하이엔드, 대중교통, 자차, 신용소비, 의료 스코어 높음 - 문화/자기소비, 여행에 대한 스코어 가장 낮음 - 지표가 가장 극명하게 나뉘는 군집
 6번	- 문화/자기소비, 여행, 재테크 스코어 높음 - 쇼핑, 대중교통, 의료 스코어 가장 낮음

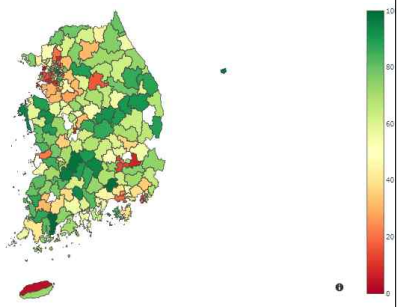
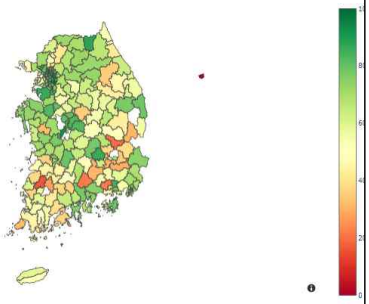
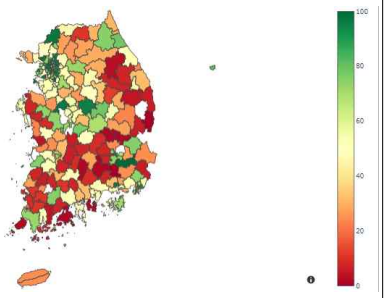
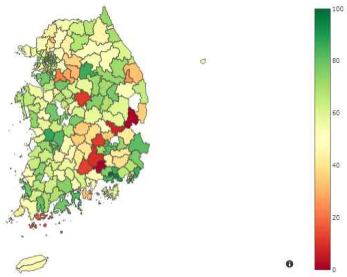
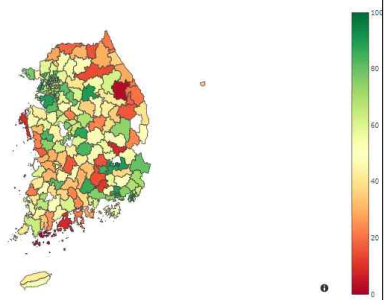
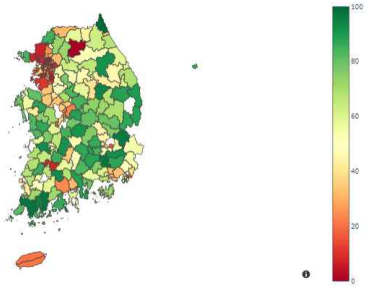
7) 지도 시각화

현재까지 진행한 지역별 클러스터링 결과를 대한민국 지도에 시각화하여 표현하였다. 0번 클러스터부터 6번 클러스터 까지 7개의 클러스터에 대해 해당 지역별로 표시하였다.

대한민국 지도 클러스터 시각화


다음은 지역별로 본 분석에서 생성한 지표의 분포이다. 지역별로 지표의 분포가 다양하게 나타남을 알 수 있다. 해당하는 지표의 스코어가 낮으면 빨간색으로, 높으면 초록색으로 나타내었다.

car_score	culture_self_score
	
customer_credit_score	digital_score
	

finanace_interest_score	high_end_score
	
medical_interest_score	public_transportation_score
	
shopping_score	travel_score
	

5. 아이디어 제안 및 기대효과

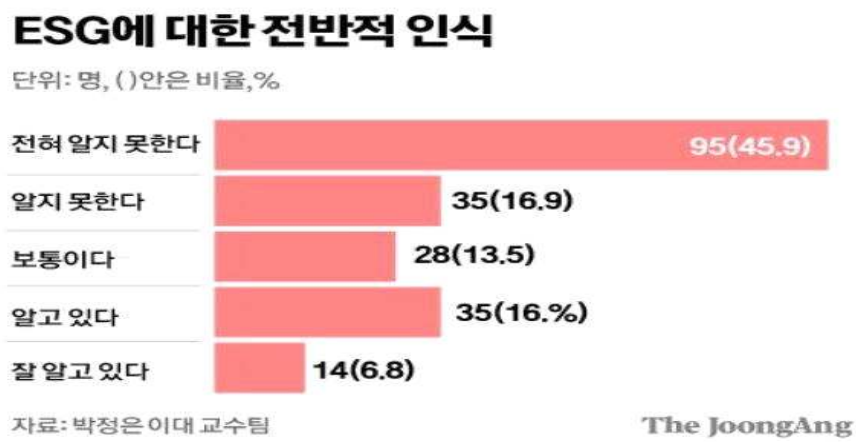
위의 방법으로 생성한 새로운 가계기반 소비지표를 통해 지역별 특징을 파악하고 segmentation을 진행할 경우 다음과 같은 새로운 서비스 상품을 제안할 수 있다.

대표적인 상품으로는 그린카드를 들 수 있다. 시중에 이미 대중교통을 이용하거나 녹색소비 생활을 할 경우 혜택을 주는 그린카드가 출시되어있긴 하지만, 이는 전국의 모든 사람에게 동일한 혜택을 제공한다. 그러나 클러스터링 분석 결과를 바탕으로 [‘대중교통’, ‘자차’, ‘여행’]등에 대한 소비지표 스코어가 비교적 높게 나타난 0번 클러스터에 속하는 지역들에게는 자동차 탄소 배출 관련 혜택이 집중된 카드를 제공한다면 카드 사용 만족도를 보다 높일 수 있다. 이러한 다양한 카드 혜택을 Segmentation 별로 특화하여 제공한다면 녹색생활 장려와 동시에 녹색 소비에 대한 관심도 증가와 같은 최신 트렌드에 발맞추어 카드 사용량을 늘릴 수 있을 것이라 생각한다.

또 다른 제안 서비스 상품은 그린론이 있다. 그린론(Green Loan)은 전기자동차나 신재생 에너지, 고효율에너지 등 친환경 관련 분야로 용도가 제한된 대출을 말한다. 기업이 그린론으로 자금을 조달하면 사업의 친환경성을 인정받을 수 있고, 금리 면에서도 유리한 조건으로 적용받을 수 있다. 은행 입장에서는 환경문제 해결에 기여할 수 있는 기업에 투자함으로써 사회적 책임을 실천할 수 있어 세계적으로 증가하는 추세이다. 이 그린론을 기업이 아닌 개인에게 적용한다면 개인의 신용 등을 고려하여 대출을 해주는 대신 친환경 관련 분야로 용도를 제한할 수 있다. 예를 들어 [‘자차’, ‘신용정도’]의 소비지표 스코어가 높게나온 4번 클러스터와 5번 클러스터에 해당하는 지역들에 대해 매연기관 자동차 대신 전기차를 구매하면 우대 금리를 제공하는 방식으로 진행한다면 환경 보존에 이바지할 수 있을 것이다.

이를 통해 기대할 수 있는 효과는 기업의 이미지 개선이다. 전국경제인연합회가 매출액 상위 500개 기업 ESG 관련 실무자를 대상으로 진행한 ‘ESG 준비실태 및 인식조사’ 결과에 따르면 43.2%가 기업 이미지 제고를 위해 ESG 경영을 도입했다고 답했다. 또한 ESG 경영이 실제 매출에 영향을 끼칠 것으로 보는 지에 대한 질문에 33.7%가 ‘차이 없다.’, 43.6%는 ’ 약간의 차이가 존재한다‘라고 대답했다. 이는 곧 기업의 입장에서 ESG 경영 도입은 기업 운영에 실질적으로 영향을 끼치지는 않지만 기업 이미지 관리를 위해 필요하다고 보는 관점을 뒷받침하는 근거로 볼 수 있다. ESG에 대한 최고경영진의 관심도는 66.3%(매우 높다 36.6%, 다소 높다 29.7%)로 높게 나타났으며, 재계 총수들을 비롯한 대다수의 CEO들은 사회 구성원과 이해관계자들의 신뢰를 받는 기업이 되기 위해 ESG 경영의 중요성을 강조하며

2021년 신년사에서 주요 키워드로 ESG 경영을 꼽기도 하였다. 그러나 많은 기업들의 경쟁적인 ESG 경영 강조가 무색하게도 정작 소비자들은 제대로 그 의미조차 인식하지 못하는 것으로 나타났다. 이화여대 경영학부의 박정은 교수와 곽윤주 연구원이 실시한 '기업의 ESG 활동에 관한 소비자의 인식과 소비자의 신뢰와 행동 의도에 미치는 영향' 연구 결과에 따르면 응답자의 62.8%(전혀 알지 못한다 45.9%, 알지 못한다 16.9%)가 ESG 경영의 의미를 잘 모른다고 응답했다.



이를 통해 ESG에 대한 적극적인 홍보를 통해 ESG의 중요성을 알려 고객 참여를 유도해야함을 알 수 있다. 우리가 제안하는 ESG 서비스의 기초가 되는 지표는 정부나 기업이 아닌 가계의 라이프스타일에 초점을 맞춰 만들었기 때문에 기존의 소비활동 지표보다 훨씬 개인에게 직접적인 영향을 끼칠 것이다. 이에 따라 우리가 제안하는 ESG 기반 서비스 또한 개인의 체감률이 훨씬 높아질 것이므로 기업의 이미지 또한 빠르게 좋은 방향으로 제고될 수 있다. 기업의 이미지 제고는 곧 국내외 수익 제고와도 직결되기 때문에 장기적으로는 기업 수익을 높이는 효과를 기대할 수도 있다. 더 나아가 해당 지역의 부족한 점을 개선할 여지가 높아지므로 지역의 이미지도 함께 제고될 수 있다.

다음으로 기대할 수 있는 효과는 환경 개선이다. ESG의 핵심 요소 중 소비자들이 가장 많은 관심을 보이는 요소는 환경(Environment)이다. 최근 기후변화보다 기후 위기라는 말이 더 어울릴 정도로 지구의 평균 온도는 점점 올라가고 각종 재해가 발생하여 전 세계적으로 기후환경과 관련된 이슈가 화제가 되고 있다. 본 분석에서 제안하는 새로운 지표를 통해 라이프스타일 및 소비성향에 맞게 세분화된 지역에

친환경, 탈탄소 사회, 그린 뉴딜, 탄소중립과 같은 핵심 키워드와 관련된 서비스를 맞춤형으로 제시한다면 좀 더 나은 환경 개선 효과를 기대할 수 있다.

6. 참고자료

- https://kosis.kr/regionState/

- 황미진(2021), '2021년도 소비자정책지표 생산 현황 및 활용 방안: 소비자생활지표를 중심으로', 한국소비자원

- https://www.j-kosham.or.kr/upload/pdf/KOSHAM-2020-20-6-389.pdf

- https://futurechosun.com/archives/55150

- https://dcollection.ewha.ac.kr/public_resource/pdf/00000184820_20220810232329.pdf