Spain

고객 세그멘테이션 분석

분석 순서

Spain의 군집 목적

Spain CRM을 만든다.

CRM column 별로 1차 군집화

군집화 정도를 보고 가공, 2차 군집화

해석의 영역----------

Spain CRM을 통해 그들의 소비 패턴을 파악----

Gold 회원, VIP, VVIP 분류 후 군집 진행

2. 전처리

0이 아닌 값들 제외

```
56]: retail_df = retail_df[retail_df['Quantity'] > 0]
retail_df = retail_df[retail_df['UnitPrice'] > 0]
retail_df = retail_df[retail_df['CustomerID'].notnull()]
print(retail_df.shape)
retail_df.isnull().sum()
```

```
(2484, 8)
```

```
56]: InvoiceNo      0
StockCode      0
Description    0
Quantity       0
InvoiceDate    0
UnitPrice      0
CustomerID     0
Country        0
dtype: int64
```

고객 당 얼마 구매했는지 누적 = Total_amount

```
In [16]: t_amount = result['Quantity'] * result['UnitPrice']
result['Total_amount'] = t_amount
result
```

Out[16]:

	CustomerID	InvoiceNo	StockCode	Quantity	InvoiceDate	UnitPrice	Total_amount
0	12557.0	536944	22383	70	2010-12-03 12:20:00	1.65	115.50
1	12557.0	536944	22384	100	2010-12-03 12:20:00	1.45	145.00
2	12557.0	536944	20727	60	2010-12-03 12:20:00	1.65	99.00
3	12557.0	536944	20725	70	2010-12-03 12:20:00	1.65	115.50
4	12557.0	536944	20728	100	2010-12-03 12:20:00	1.45	145.00
...	...	...	...	...	...	...	...
144	12442.0	580955	21974	12	2011-12-06 14:22:00	1.45	17.40
145	12442.0	580955	23597	6	2011-12-06 14:22:00	2.95	17.70
146	12442.0	580955	22090	6	2011-12-06 14:22:00	2.95	17.70
147	12442.0	580955	21209	12	2011-12-06 14:22:00	0.39	4.68
148	12442.0	580955	21981	24	2011-12-06 14:22:00	0.39	9.36

149 rows × 7 columns

분석 들어가기 전에

적은 고객으로 인한 분포 확인이 어렵습니다. 미리 양해의 말씀을 구합니다 ^^

고객 31명

```
In [54]: retail_df['CustomerID'].unique()

Out[54]: array([12557., 17097., 12540., 12551., 12503., 12484., 12539., 12510.,
                12421., 12502., 12462., 12507., 12541., 12547., 12597., 12545.,
                12596., 12354., 12417., 12455., 12450., 12548., 12556., 12550.,
                12546., 12454., 12448., 12544., 12538., 12445., 12442.])
```

** 고객 별 소비 금액 등급 나눔 (target)

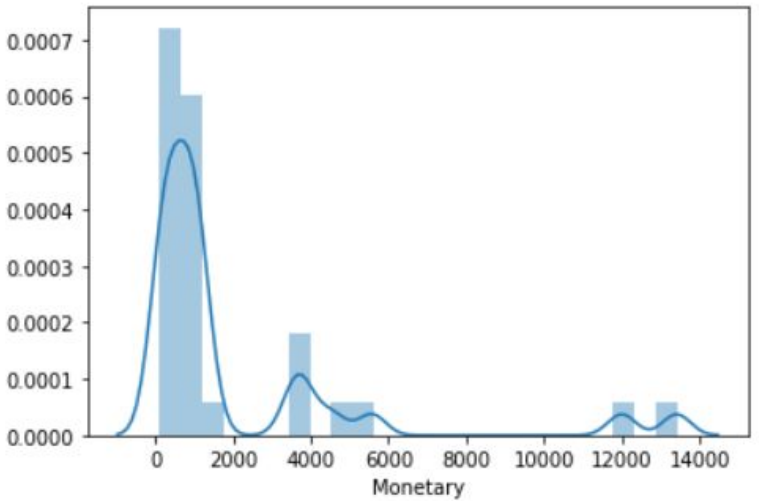
Silver, Gold, Diamond 회원으로 나눔

값이 편중되어 있어 로그를 써서 진행

고객 별 소비별 금액에 따른 등급 나눔

```
sns.distplot(cust_df['Total_amount'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x282746d66c8>
```

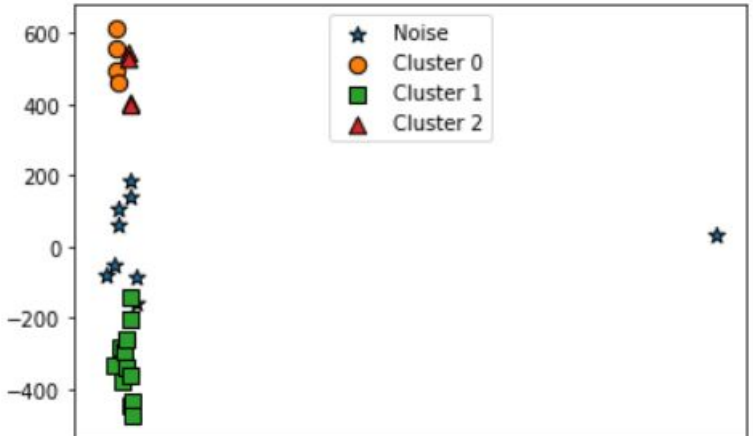


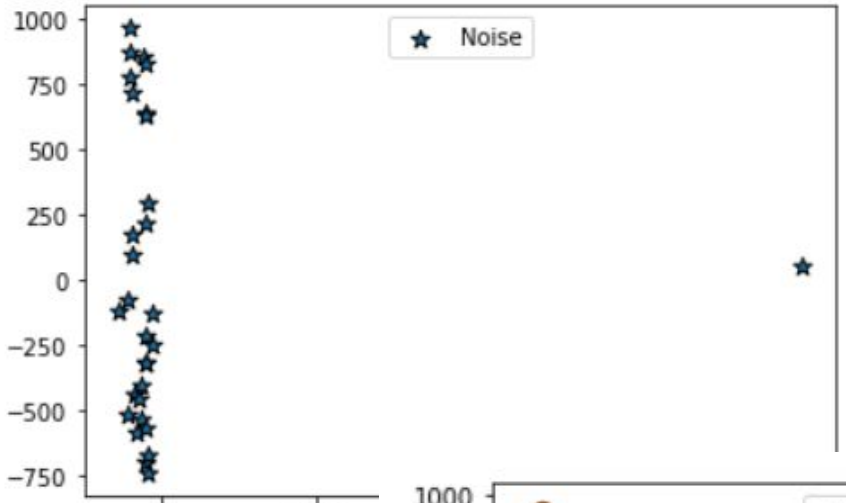
3. 1차 - DBSCAN 에 따른 군집화
파라미터 설정에 따라 노이즈 확인

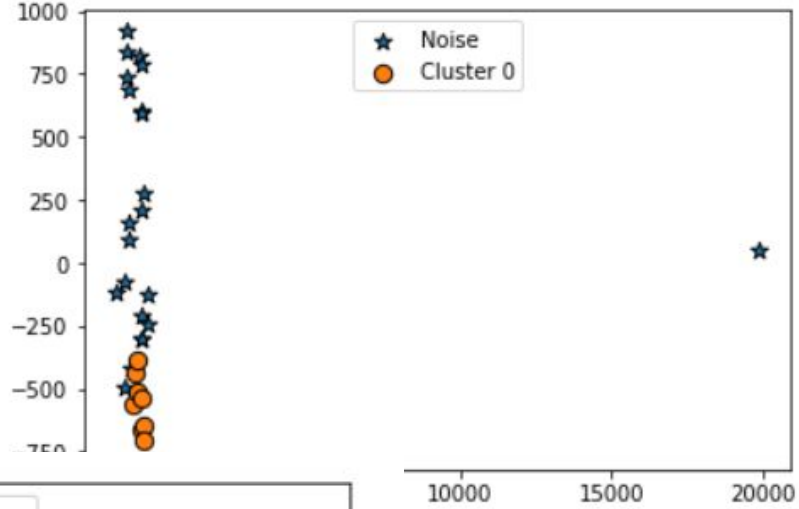
35]:

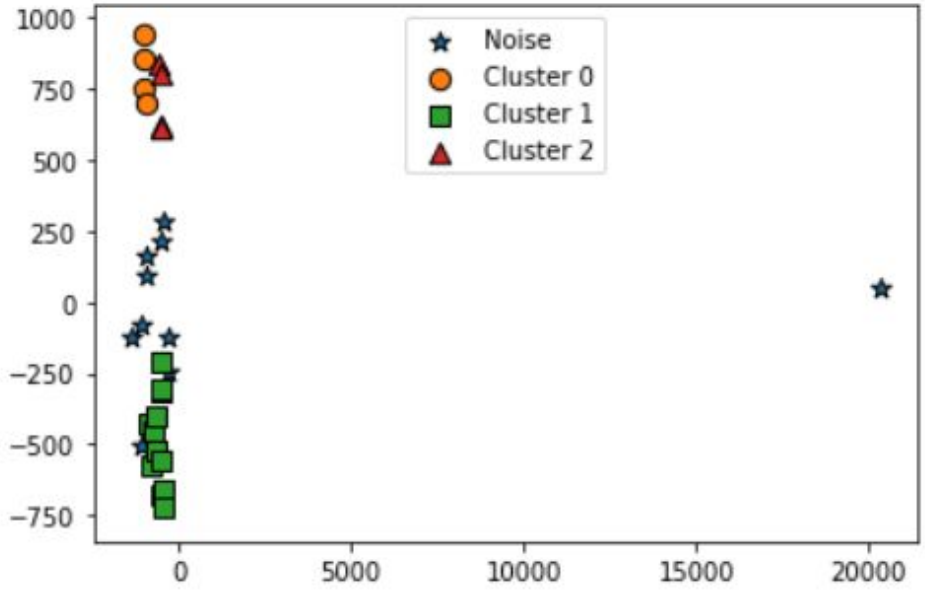
```
# 2차원으로 시각화하기 위해 PCA n_componets=2로 피처 데이터 세트 변환
pca = PCA(n_components=2, random_state=0)
pca_transformed = pca.fit_transform(cus_df2)
# visualize_cluster_2d( ) 함수는 ftr1, ftr2 컬럼을 좌표에 표현하므로 PCA 변환값을 해당 컬럼으로 생성
cus_df2['ftr1'] = pca_transformed[:,0]
cus_df2['ftr2'] = pca_transformed[:,1]

visualize_cluster_plot(dbscan, cus_df2, 'dbscan_cluster', iscenter=False)
#eps = 150, min_samples = 4
```




Noise
1000
750
500
250
0
−250
−500
−750
0
5000


Noise
Cluster 0
1000
750
500
250
0
−250
−500
10000
15000
20000


Noise
Cluster 0
Cluster 1
Cluster 2
1000
750
500
250
0
−250
−500
−750
0
5000
10000
15000
20000

5. 실 데이터와 확인하여 노이즈 여부 결정

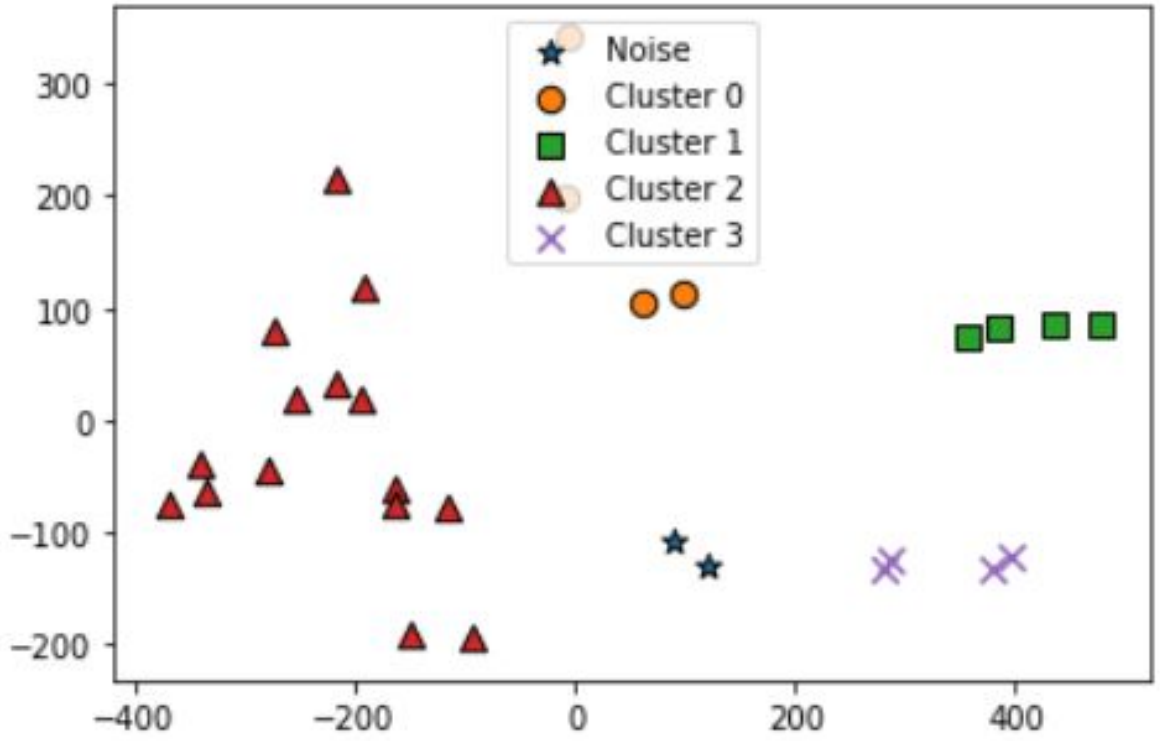
DBSCAN을 통한 군집화를 진행했는데,

노이즈로 인해 시각화하는데 어려움이 있었다.

노이즈 제거 후에 다시 DBSCAN 진행.

2차 - DBSCAN 에 따른 군집화

→ 명확한 군집 형성



6. 각 군집마다 어떤 고객 군인지 확인

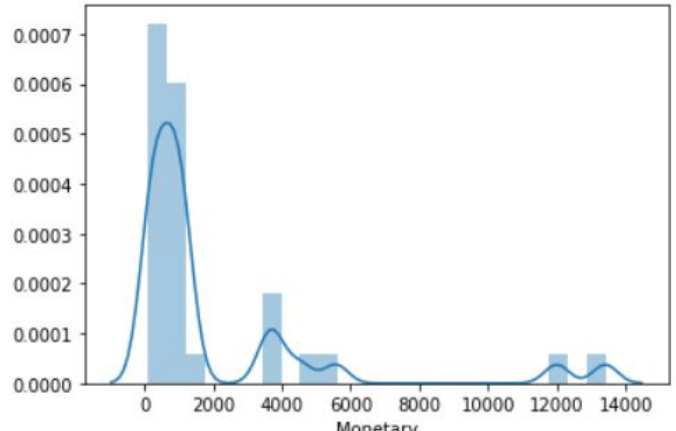
★기존에 소비누적 그래프에서 분산된 부분 = 다이아몬드 회원

앞서 노이즈는 다이아몬드 회원이라고 추정 → 적중 48.2%

고객 별 소비별 금액에 따른 등급 나눔

```
sns.distplot(cust_df['Total_amount'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x282746d66c8>
```




```
dfff[dfff['Grade']==dfff['dbscan_cluster']].count()
#11개
```

```
CustomerID        11
InvoiceDate       11
InvoiceNo         11
Quantity          11
Total_amount      11
Grade             11
dbscan_cluster    11
ftr1              11
ftr2              11
dtype: int64
```

```
print('군집 확인 후 적중', round(14/ 29 * 100),2, '%')
```

```
군집 확인 후 적중 48 2 %
```

-END-