HW

짱짱조

[1] Dataset and Pre-Processing

1. dataset

Opendatasoft 에 공개된 에어비앤비(Airbnb) 숙소 리스트 데이터를 활용하였다. 전 세계 도시에서 실제로 Airbnb 가 어떻게 사용되는지 알 수 있는 숙소 리스트 데이터이다. 공개된 494,954 개의 데이터 중 **미국 뉴욕시에 등록된 10,000 개의 숙소 등록 정보를 API 를 통해 받아와 데이터셋으로 사용하였다**.

데이터셋에는 숙소 정보(침대 개수, 구비 물품, 방 개수 등), 호스트 정보(연락 수단, 슈퍼호스트 인증 등), 예약 조건(예약 가능 인원, 가격, 숙박 가능 기간 등), 리뷰(청결도, 의사소통, 위치 등)에 대한 다양한 변수가 포함된 정보가 담겨 있다. 그 중 우리는 연속형 변수이면서 지난 FA 실습에서 유의미한 결과를 보여주었던 변수들인 숙소 당 월별 리뷰 수의 평균치, 숙소 가격, 연중 이용가능도를 각 변수로 삼아 clustering 을 진행하였다.

전처리를 진행하기 전 결측치만 제거한 데이터셋에 기초 통계량은 다음과 같았다.

	reviews_per_month	price	availability_365
count	8306.000000	8306.000000	8306.000000
mean	1.905714	141.237900	178.875271
std	1.871497	115.662734	135.155154
min	0.010000	10.000000	0.000000
25%	0.482500	72.000000	45.000000
50%	1.270000	105.000000	165.500000
75%	2.760000	170.000000	320.000000
max	12.630000	999.000000	365.000000

2. preprocessing

전처리는 outlier 을 제외한 후, 정규화를 해주는 방식으로 진행되었고 다음과 같은 코드를 이용하였다.

```
def get_outlier(df=None, column=None, weight=1.5):
  quantile_25 = np.percentile(df[column].values, 25)
  quantile_75 = np.percentile(df[column].values, 75)

  IQR = quantile_75 - quantile_25
  IQR_weight = IQR*weight

  lowest = quantile_25 - IQR_weight
  highest = quantile_75 + IQR_weight

  outlier_idx = df[column][ (df[column] < lowest) | (df[column] > highest) ].index
  return outlier_idx
```

```
data = pd.DataFrame(StandardScaler().fit_transform(data), columns=data.columns, index = data.index)
data.head()
```

그 결과 전처리를 진행한 후 통계량은 다음과 같았다.

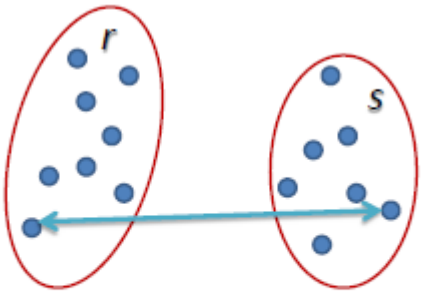
	reviews_per_month	price	availability_365
count	7.492000e+03	7.492000e+03	7.492000e+03
mean	2.215704e-16	4.012926e-17	1.519444e-16
std	1.000067e+00	1.000067e+00	1.000067e+00
min	-1.127847e+00	-1.658682e+00	-1.294843e+00
25%	-8.186951e-01	-7.466354e-01	-9.947178e-01
50%	-3.286561e-01	-2.906123e-01	-1.164377e-01
75%	5.840004e-01	4.694262e-01	1.061967e+00
max	2.943752e+00	3.144762e+00	1.393394e+00

[2] Hierarchical clustering

1. Linkage method 비교

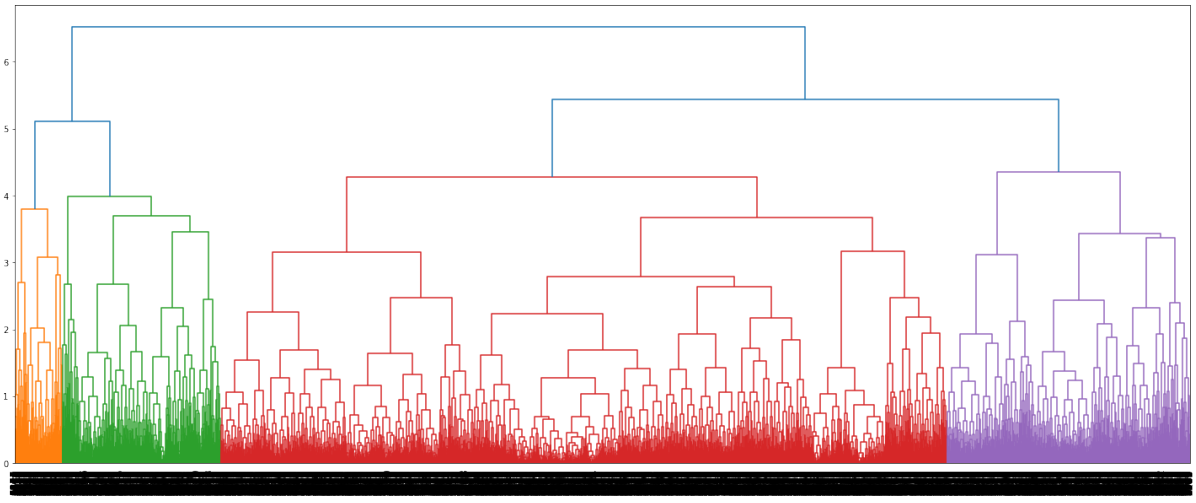
계층적 군집분석을 진행할 때 두 클러스터 간 거리를 측정하는 방법에는 single, complete, average, centroid, ward 등이 있다. 그 중 complete와 average 방법을 비교해 보았다.

1) complete

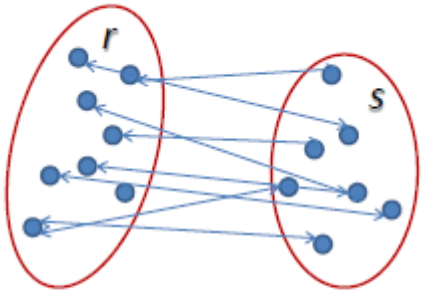


$$L(r,s) = \max(D(x_{ri}, x_{sj}))$$

complete 연결 방법은 각 클러스터에서 하나씩 뽑은 point 쌍 중 가장 멀리 떨어진 거리를 클러스터 간 거리로 정의한다. 이는 새로 형성되는 클러스터의 전체 크기를 가능한 작게 만드는 방향으로 동작하나, 내부의 밀도가 높다는 보장은 할 수 없다. complete 연결 방법으로 도출된 덴드로그램은 다음과 같다.

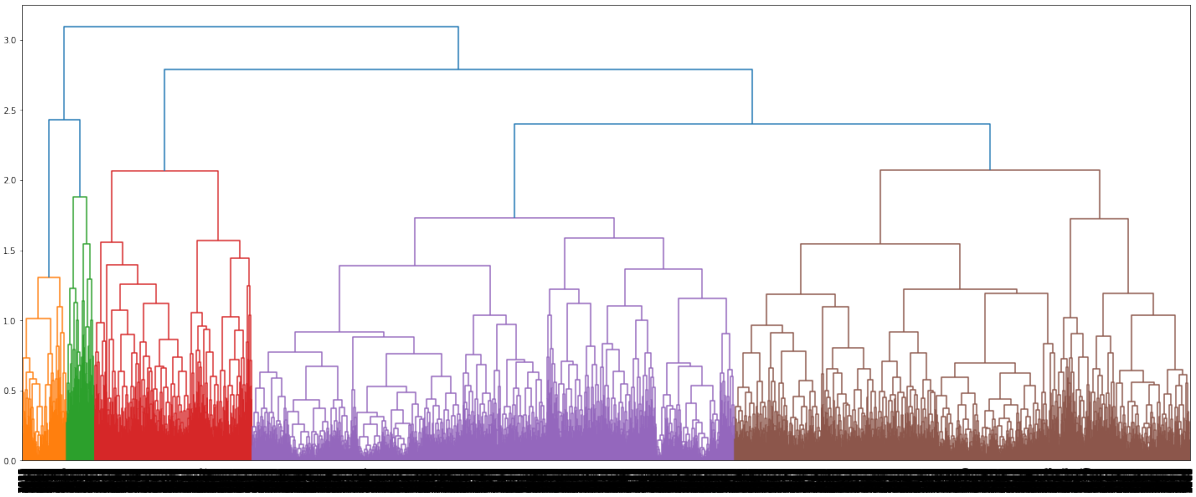


2) average



$$L(r,s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} D(x_{ri}, x_{sj})$$

average 연결 방법은 각 클러스터에서 하나씩 뽑은 point 쌍들의 평균 거리를 클러스터 간 거리로 정의한다. 이는 모든 원소의 영향을 동일하게 반영한다는 특징이 있다. average 연결 방법으로 도출된 덴드로그램은 다음과 같다.

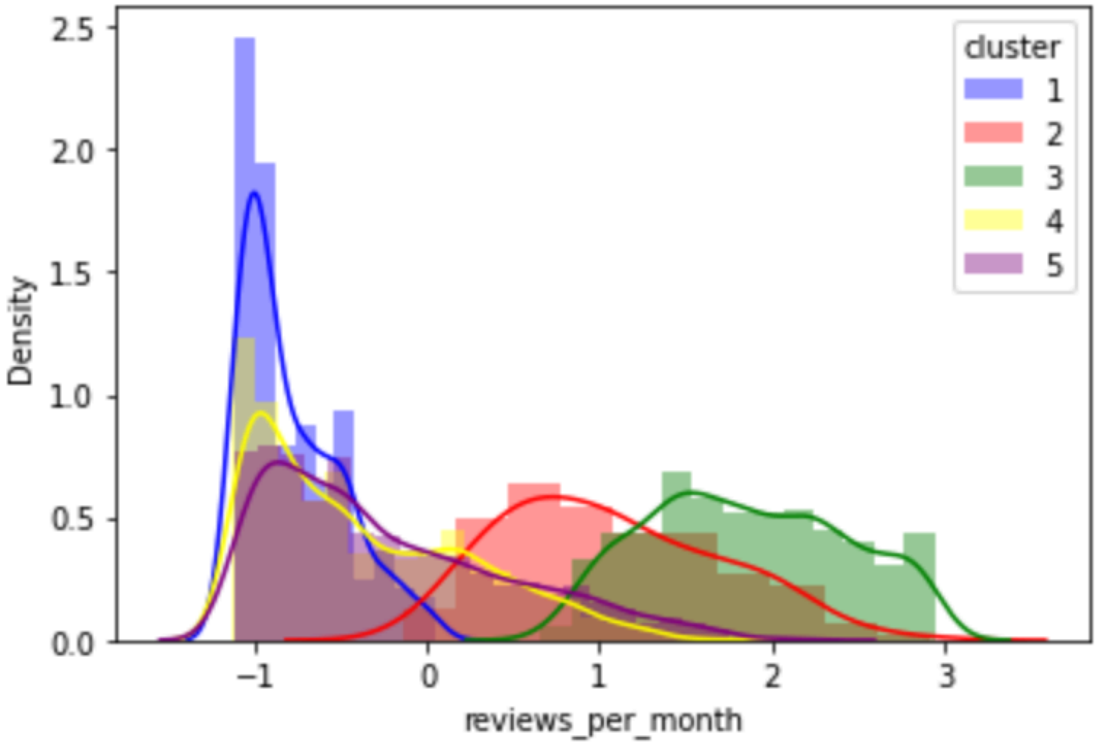


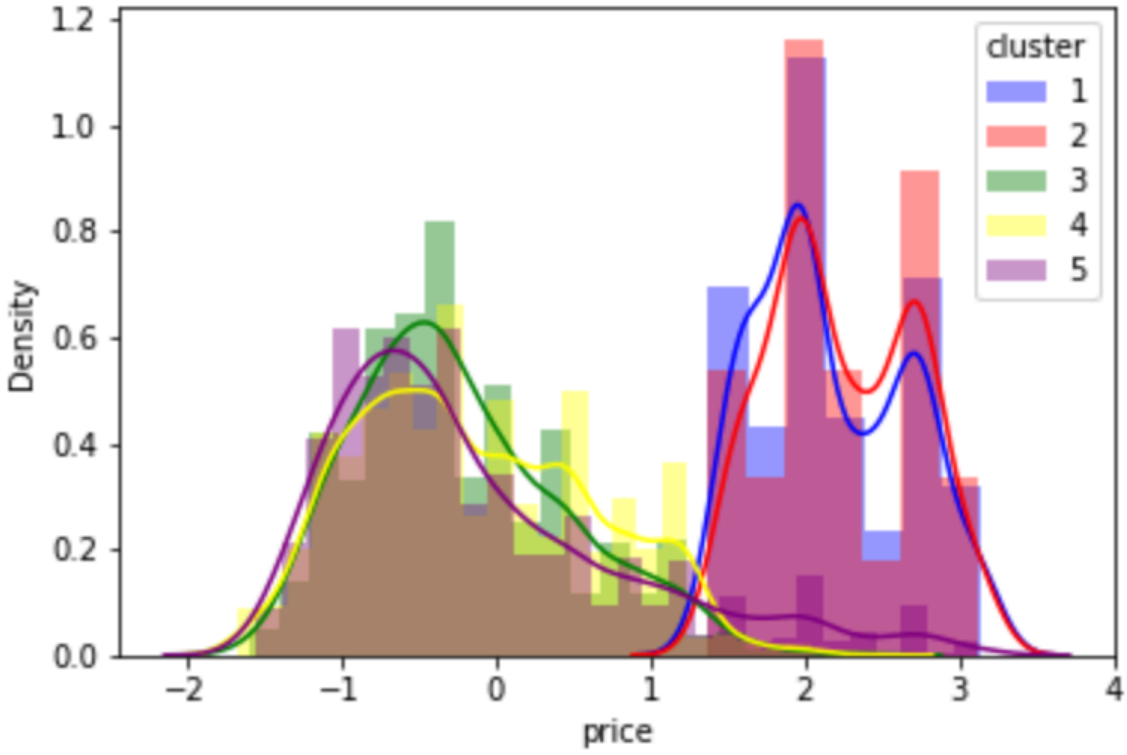
2. 결과 분석

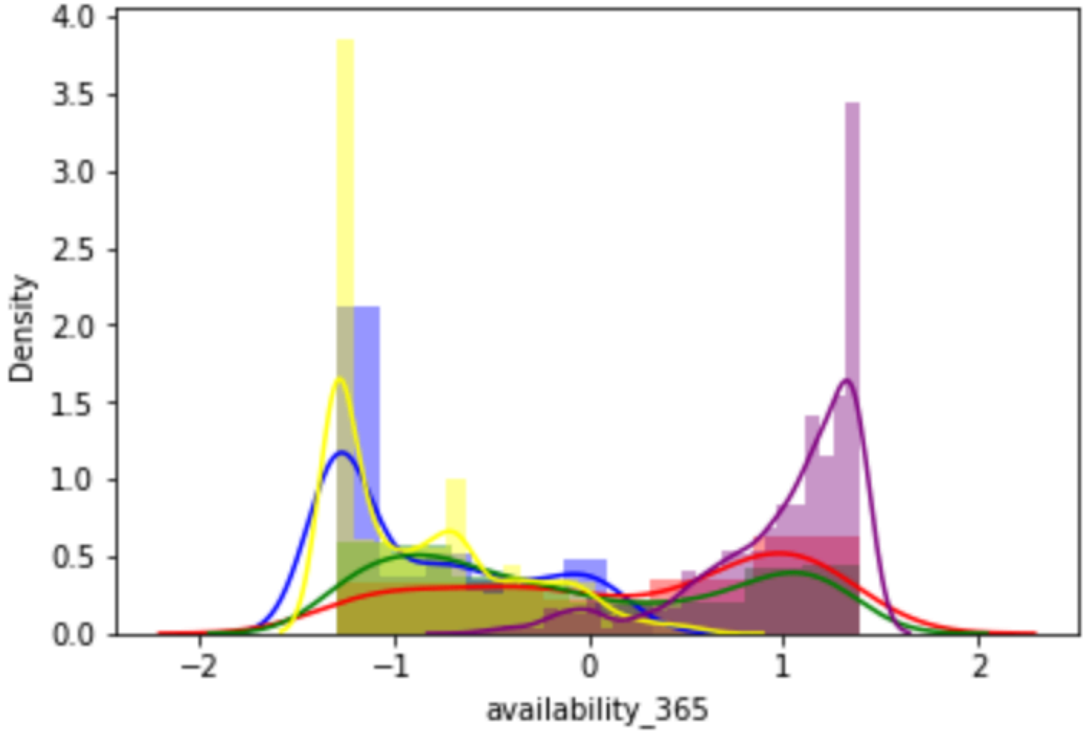
군집 수를 직접 정해서 군집 분석을 실시하는데 사용할 방식은 위에서 언급한 바와 같이 average 를 채택하였다. 해당 덴드로그램에서 색이 총 5 개로 분류되었고 군집 간 높이의 차이가 큰 부분 역시도 5 개의 군집으로 분류되어서 군집의 수를 5 로 택하기로 했다. 결과값 비교를 위해 t 값은 2.3 으로 설정해주었다. 그 결과로 도출된 5 개의 군집의 통계량은 다음과 같았다.

cluster index	reviews_per_month		price		availability_365	
	mean	std	mean	std	mean	std
1	-0.785	0.288	2.176	0.485	-0.834	0.504
2	1.062	0.634	2.244	0.462	0.207	0.849
3	1.857	0.575	-0.222	0.686	-0.090	0.863
4	-0.378	0.620	-0.136	0.762	-0.849	0.468
5	-0.230	0.699	-0.129	0.963	1.000	0.412

이를 시각화해보기 위해 다음과 같이 히스토그램을 그려보았다.




cluster
1
2
3
4
5
Density
price
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−2
−1
0
1
2
3
4


Density
availability_365
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−2
−1
0
1
2

3. 군집 분석

가. cluster 1

cluster 1 의 경우 3 개의 변수 모든 부분에서 한 쪽으로 쏠려 있음을 확인할 수 있었다. 리뷰 수는 적은 축에 속했고, 가격은 비싼 쪽에 availability 는 낮은 쪽에 속했다.

=> cluster1: 리뷰 수가 적고 가격이 비싸고 연중 이용가능도가 낮은 숙소의 군집

나. cluster 2

cluster 2 의 경우 리뷰 수의 경우에는 군집 별 데이터의 차이로 인해 히스토그램에서는 거의 평균에 위치한 것으로 보이지만, 절대적인 수치를 보면 어느정도 어느 정도 리뷰가 있는 것으로 해석된다. 또한 가격은 비싼 쪽에 속했으며 availability 에서는 이렇다할 특성이 없었다.

=> cluster2: 적당한 수의 리뷰를 보유하고 가격이 비싼 숙소의 군집

다. cluster3

cluster3 의 경우 리뷰 수는 많은 쪽에 쏠려 있으며, price 는 낮은 쪽에 쏠려 있었다. 하지만 availability 는 전반적으로 고르게 분포해 이렇다할 특성을 보여주지는 않았다.

=> cluster3: 많은 리뷰를 보유하고 가격이 싼 숙소의 군집

라. cluster4

cluster4 의 경우에는 리뷰 수는 적은 쪽에 쏠려 있었다. 이 부분에 대해서 cluster 1 과 비교했을 때 덜해 보이지만 이는 데이터 수에 의한 것이므로 충분히 리뷰 수는 적다고 해석할 수 있다. 가격과 availability 는 확실하게 낮은 쪽에 쏠려 있음을 확인할 수 있었다.

=> cluster4: 리뷰 수가 적고 가격이 싸고 연중 이용가능도가 낮은 숙소의 군집

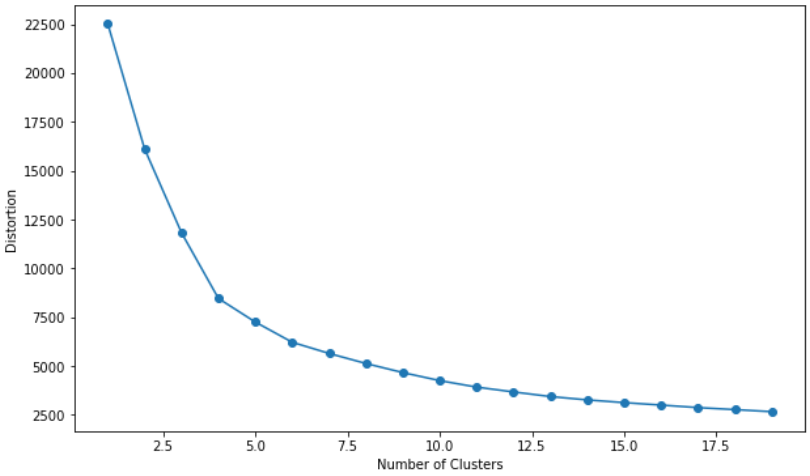
마. cluster5

cluster5 의 경우에는 리뷰 수 변수에서는 cluster4 와 같은 맥락으로 리뷰 수가 적다고 해석할 수 있었고 가격 역시 cluster4 와 비슷하게 낮았지만, 연중 이용가능도가 다른 군집과 다르게 유일하게 높은 쪽에 쏠려 있음을 확인할 수 있었다.

=> cluster5: 리뷰 수가 적고 가격이 싸고 연중 이용가능도가 높은 숙소의 군집

[3] K-Means Clustering

1. K-Means Clustering 수 선정



적절한 Clustering 수를 선정하기 위해 cluster 의 개수에 따른 데이터의 군집내 오차 제곱합을 그래프로 나타내어 Cluster 수를 선정하는 elbow 기법을 사용하였다.
클러스터의 개수가 1~20 일 때의 오차 제곱 합(SSE)을 그래프로 나타낸 결과 클러스터의 개수가 4 개일때까지 기울기가 급격하다가 그 이후부터 점차 완만해지는 것을 볼 수 있다. 이에 따라 이번 과제에서는 4 개의 군집이면 가장 효율적이고 적합하게 데이터를 군집할 수 있을 것이라는 판단으로 군집화 개수는 4 개로 진행했다.

2. Clustering 결과

1) Cluster 별 평균 및 표준편차

cluster index	reviews_per_month		price		availability_365	
	mean	std	mean	std	mean	std
0	1.617	0.625	-0.255	0.698	0.082	0.854
1	-0.340	0.549	-0.396	0.586	1.071	0.332
2	-0.314	0.712	-1.717	0.650	0.041	0.955
3	-0.496	0.515	-0.373	0.603	-0.902	0.420

[표 2] k-means 군집 별 평균(왼), 군집 별 표준편차(오)

Cluster 별로 평균과 표준편차를 분석해보았다.

① Cluster 0

- 월별 리뷰 수 (R): 전체 평균보다 높은 1.6 의 평균값으로 치우친 분포를 보이며 4 개의 군집 중 유일하게 양의 값을 갖는다
- 가격(P): -0.25 의 평균값으로 어느정도 치우쳐 있는 분포를 가진다.
- 이용가능 일수(A): 평균이 0.08 로 0 에 거의 가까우며 표준편차도 매우 높아 치우침없이 고르게 분포를 가진다.

② Cluster 1

- 월별 리뷰 수: -0.34 의 평균값으로 어느정도 치우쳐 있는 분포를 가진다.
- 가격: -0.396 의 평균과 0.586 의 표준편차로 전체평균보다 낮은 값에 어느정도 치우친 분포를 가진다.
- 이용가능 일수: 1.07 의 평균과 0.33 의 표준편차를 가지며 전체평균보다 높은 값에 매우 치우친 분포를 보인다.

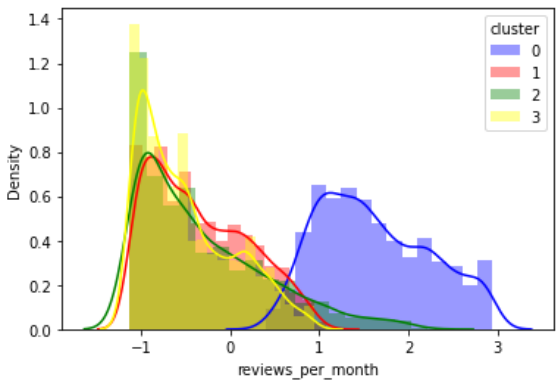
③ Cluster 3

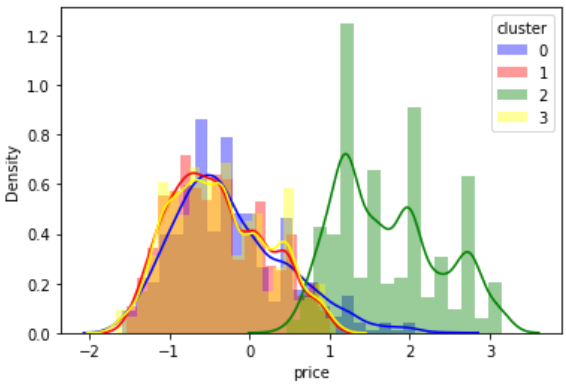
- 월별 리뷰 수: -0.315 의 평균과 0.712 의 표준편차로 전체 평균보다 낮은 값에 어느정도 치우친 분포를 가진다.
- 가격: 1.717 의 평균과 0.650 의 표준편차로 전체 평균보다 낮은 값에 어느정도 치우친 분포를 가진다.
- 이용가능 일수: 평균이 0.041 로 0 에 거의 가까우며 표준편차도 매우 높아 치우침 없이 고르게 분포를 가진다.

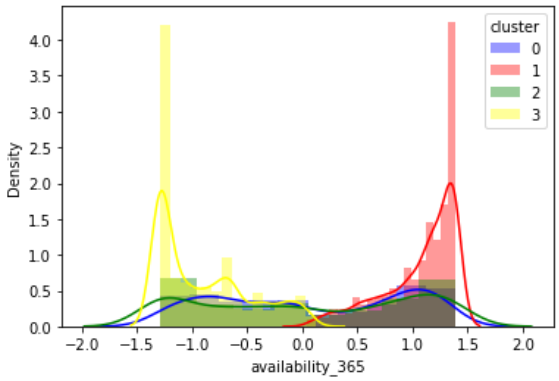
④ Cluster 4

- 월별 리뷰 수: -0.496 의 평균과 0.515 의 표준편차로 전체 평균보다 낮은 값에 매우 치우친 분포를 가진다.
- 가격: -0.373 의 평균과 0.603 의 표준편차로 전체 평균보다 낮은 값에 어느정도 치우친 분포를 가진다.
- 이용가능 일수: -0.902 의 평균과 비교적 낮은 0.420 의 표준편차를 가지며 전체평균보다 낮은 값에 매우 치우친 분포를 보인다.

2) Cluster 별 분포 시각화







군집 별 평균, 표준편차, 시각화 된 분포를 모두 종합했을 때 각 군집 별 특징은 다음과 같다.

- Cluster0: 가격이 싼 경향(-P)을 보이는 방
- Cluster1: 이용가능 일수가 매우 많은 경향(A)을 보이는 방
- Cluster2: 가격이 비싼 경향(P)을 보이는 방
- Cluster3: 월별 리뷰 수와 이용가능 일수가 매우 적은 경향(-R, -A)을 보이는 방

[4] Hierarchical Clustering 과 K-Means Clustering 비교

Hierarchical Clustering 한 결과 전체 데이터셋에 대한 Silhouette Analysis Score 는 0.411 이 나왔고 각 군집별로는 1,2,3,4,5 번 군집 각각 0.399, 0.446, 0.313, 0.481, 0.436 이 나왔다. 군집 중에서는 4 번 군집이 가장 성능이 좋은 것을 확인할 수 있다.

K-means Clustering 한 결과 전체 데이터셋에 대한 Silhouette Analysis Score 는 0.484 가 나왔고 각 군집별로는 0,1,2,3 번 군집 각각 0.371, 0.549, 0.329, 0.575 가 나왔다. 군집 중에서는 2,3 번 군집이 가장 성능이 좋은 것을 확인할 수 있다.

종합하여, 본 데이터셋에서는 k-means clustering 이 hierarchical clustering 보다 유의미한 결과를 도출하였다.

[5] Reference

- 데이터셋
https://public.opendatasoft.com/explore/dataset/airbnb-listings/table/?disjunctive.host_verifications&disjunctive.amenities&disjunctive.features