L사의 고객 세분화를 통한 맞춤형 상품 및 서비스 추천

분석프로그래밍 프로젝트 - 분석보고서

빅데이터경영통계전공 20192761 김정하

1. DATA CLEANSING

- 결측값 확인 및 정리
- 데이터 타입 변환
- 데이터 둘러보기

01

DATA CLEANSING

```
cs = pd.read_csv('L사_고객정보.csv')
gd = pd.read_csv('L사_상품정보.csv')
tr = pd.read_csv('L사_거래정보.csv')
log = pd.read_csv('L사_로그정보.csv')
```

1. 결측값 확인 및 정리, 데이터 타입 변환 - cs, gd, tr, log

1.1 고객정보, cs

- 결측값은 존재하지 않으나, 성별과 연령 column에서 “unknown”값이 존재함.
  하지만 cs, gd, tr을 merge하면 “unknown”으로 되어있는 고객이 존재하지 않음.
  즉, tr에서 거래정보가 존재하지 않는 고객은 성별, 연령이 “unknown”

1.2 상품정보, gd

-cs,tr,gd merge를 위해 gd의 pd_c 열의 데이터 유형과 자리수를 맞춰줌.

1.3 거래정보, tr

- 결측값은 존재하지 않음.
- de_dt열의 데이터 유형을 int에서 datetime으로 변경해줌.

```
In [217]: log.isnull().sum()
Out[217]: clnt_id              0
          sess_id              0
          hit_seq              0
          action_type          0
          biz_unit             0
          sess_dt              0
          hit_tm               0
          hit_pss_tm           0
          trans_id       3139373
          sech_kwd       2544724
          tot_pag_view_ct     1428
          tot_sess_hr_v      57607
          trfc_src               0
          dvc_ctg_nm       1782577
          dtype: int64

action_type이 6(구매 완료), 7(구매 환불)이 아닌 경우는 trans_id가 결측값

In [218]: log.query("action_type in [0,1,2,3,4,5,8]").trans_id.unique()
Out[218]: array([nan])

action_type이 0(검색)이 아닌 경우는 sech_kwd가 결측값

In [219]: log.query("action_type in [1,2,3,4,5,6,7,8]").sech_kwd.unique()
Out[219]: array([nan], dtype=object)
```

```
tot_pag_view_ct와 tot_sess_hr_v는 action_type과 관계없이 결측값 발생

In [220]: log.query("action_type in [5]").tot_pag_view_ct.isna().value_counts()
Out[220]: False    750459
          True         71
          Name: tot_pag_view_ct, dtype: int64

In [221]: log.query("action_type in [0]").tot_sess_hr_v.isna().value_counts()
Out[221]: False    649342
          True       2296
          Name: tot_sess_hr_v, dtype: int64

dvc_ctg_nm는 action_type이 8 (결제 옵션)일때를 제외하고 결측값 발생

In [222]: log.query("action_type in [8]").dvc_ctg_nm.unique()
Out[222]: array([], dtype=object)

sess_dt 열의 데이터 유형을 int에서 datetime으로 변경

In [223]: log.sess_dt = log.sess_dt.astype(str).astype('datetime64')
```

1.4 로그정보 , log

- action_type에 따라 결측값 발생.
  action_type이 구매 완료, 구매 환불이 아닌 경우, trans_id가 결측값.
  검색이 아닌 경우 sech_kwd가 결측값.
  결제 옵션이 아닐 경우 dvc_ctg_nm이 결측값.
  tot_pag_view_ct 와 tot_sess_hr_v는 action_type과 관계없이 결측값.
  따라서, 다른 데이터들과 merge하는 것보다는 따로 활용하는게 더 좋다고 생각함.
- sess_dt열의 데이터 유형을 int에서 datetime으로 변경해줌.

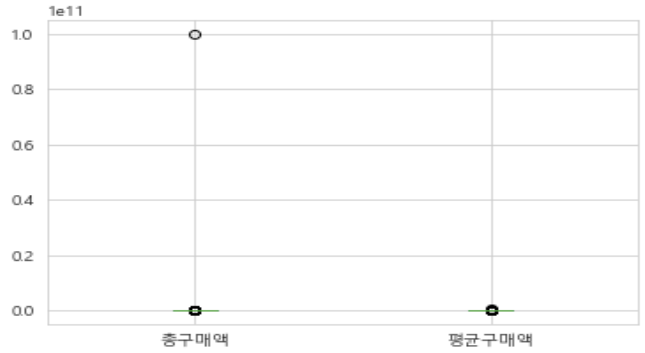
01

DATA CLEANSING

1. 결측값 확인 및 정리, 데이터 타입 변환 - md

1.5 MD (cs, tr, gd를 merge한 데이터)

- boxplot을 통해 이상치를 확인하여 구매액이 과도하게 큰 고객을 제거함.



buy_am, buy_ct 열 정리하기

```
# 둘 다 0인 경우 삭제
md = md.drop(md.query(" buy_am ==0 and buy_ct == 0").index)

# 금액은 0이 아니지만 수량이 0인 경우
# 구매금액이 평균보다 크면 수량을 2, 작으면 1로 변경
md['buy_ct'] = np.where(md['buy_am'] > md['buy_am'].mean(), 2 , 1 )

# 수량은 0이 아니지만 금액이 0인 경우는 전체의 0.01%이므로 삭제
md.query("buy_am == 0 and buy_ct != 0 ").shape[0] / md.shape[0]
md = md.drop(md.query(" buy_am ==0 and buy_ct != 0").index)
```

- buy_am과 buy_ct열에 이상이 있는 경우 3가지

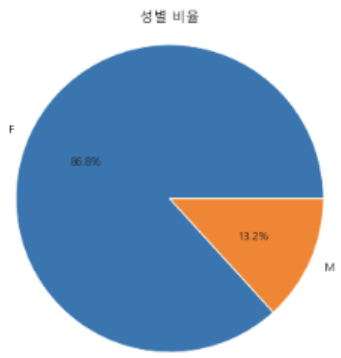
① buy_am과 buy_ct가 0인 경우, 값을 채울 때 참고할 대상이 없으므로 삭제.

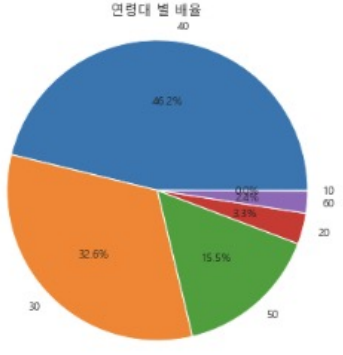
② buy_am != 0 , buy_ct ==0 인 경우,
buy_am이 평균보다 크면 buy_ct를 2, 평균보다 작으면 1로 채워줌.

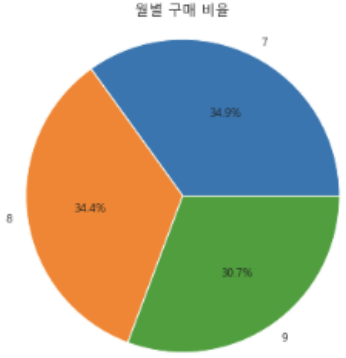
③ buy_am == 0 , buy_ct !=0 인 경우는 전체 데이터의 0.01%에 해당함으로 삭제.

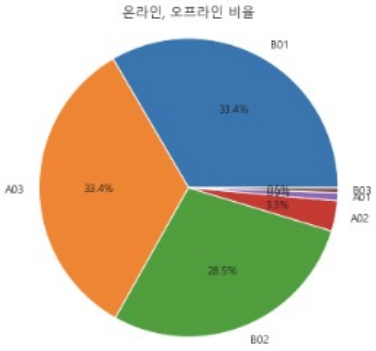
2. MD 데이터 간단하게 살펴보기

- 성별 비율이 F(여성)가 월등히 높고,
연령대는 40대, 30대가 80%가까이 차지함.
월별 구매 비율은 7,8,9월이 비슷하게 나타나고,
오프라인 구매비율이 더 높게 나옴.









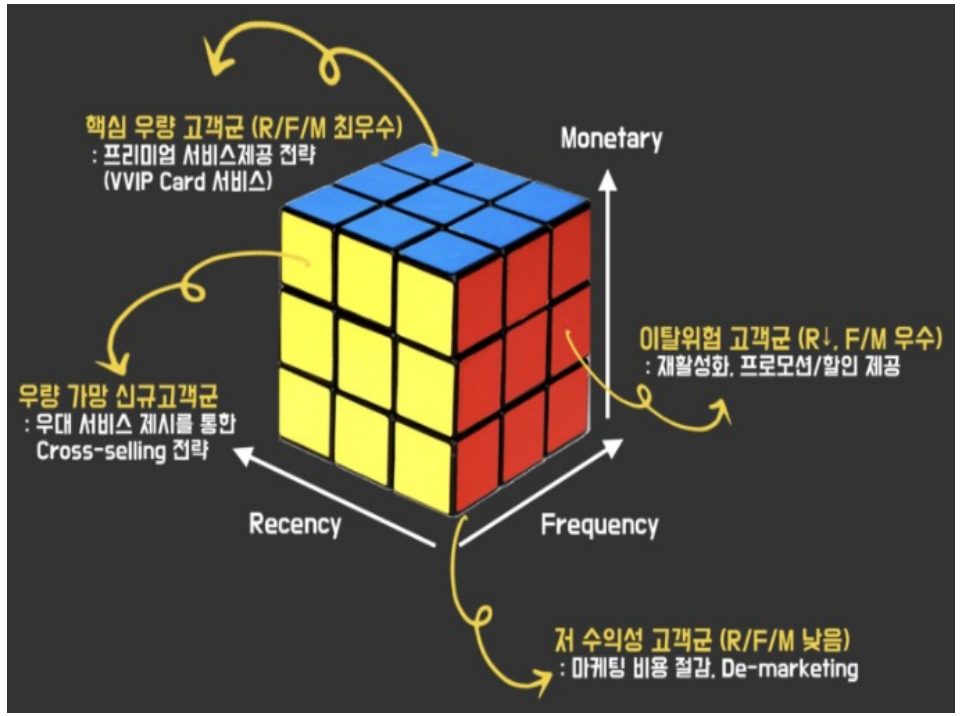
2. RFM 분석

- RFM 분석 선택 이유
- recency, frequency, monetary
- RFMgroup에 따른 RFM_segment

02

RFM 분석

1. RFM 분석을 선택한 이유



RFM 분석은 고객의 최근성(R), 구매빈도(F), 구매액(M)
이 3가지의 척도를 통해
해당 고객이 기업에게 얼만큼의 수익을 가져다주는지 판단할 수 있도록 함.
기업의 입장에서 생각해보았을때,
RFM 분석을 통해 고객을 핵심 우량 고객군, 우량 가망 신규고객군,
저 수익성 고객군, 이탈위험 고객군으로 세분화하여
어떤 고객이 앞으로의 구매 가능성이 높은지,
적절한 마케팅 방법과 그 타켓이 누구인지를 판단하는 것이
고객 세분화에 있어 가장 중요할 것이라 생각되어 RFM 분석을 선택.

02

RFM 분석

2. RFM 분석 - R , F , M

2.1 recency

-rfm 분석에서 R은

해당 고객이 이탈고객인지 아닌지를 판단하는 척도가 되는데,

이때 md 데이터만 활용하여 recency를 구한다면

최근 구매이력은 없지만 온라인 접속을 한 기록이 있는 사람도

이탈고객으로 분류됨.

- 따라서 온라인 접속 이력까지 포함하여 recency를 구해야만

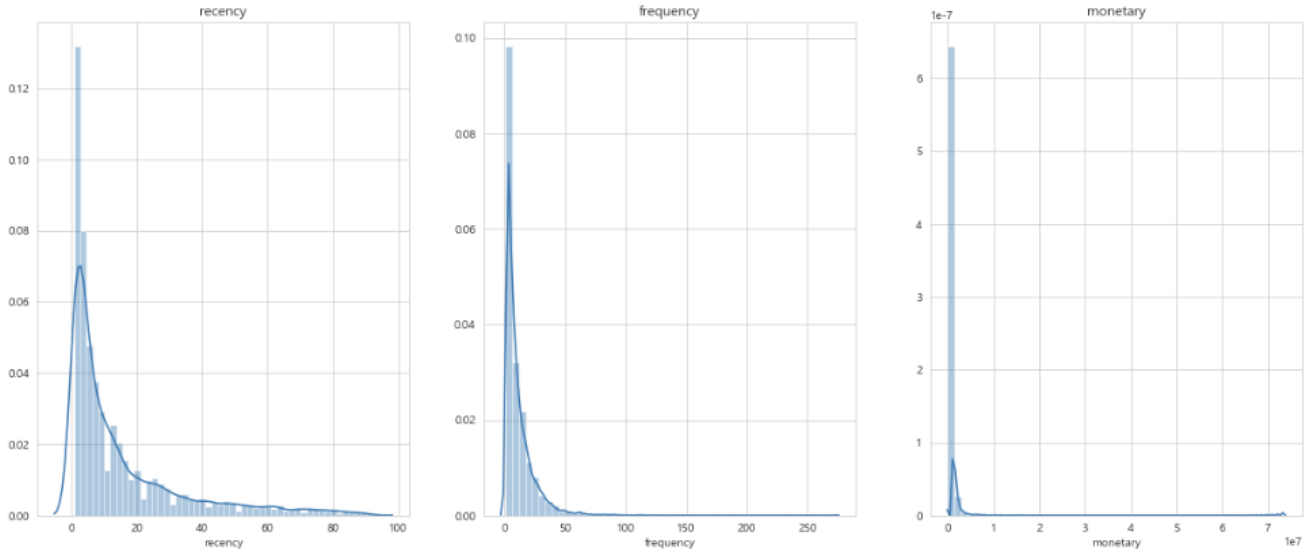
그 고객의 이탈유무를 정확하게 파악할 수 있음.

2.2 fequency

- 고객별 transid를 unique로 센 값을 frequency로 저장함.

2.3 monetary

- 고객별 총구매액을 monetary로 저장함.



2.4 recency, frquency, monetary별 displot

- recency와 frequency의 경우, 비교적 넓게 분포하고 있음.
- monetary는 한쪽으로 치우쳐 집중되어 있음을 확인할 수 있음.

이는 monetary가 recency와 frquency에 비해

고객을 세분화하기에 좋은 척도가 아님을 의미함.

따라서 총구매액이 아닌 금액과 관련한 다른 여러 feature들을 만들어

더 자세하게 segmnet를 나눌 필요성이 있다고 생각했음.

02

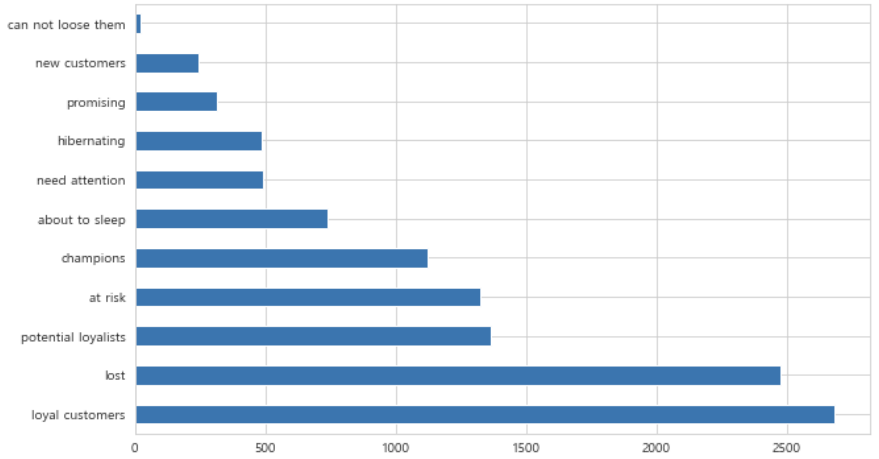
RFM 분석

3. RFM 분석 - RFMgroup에 따른 RFM_segment 지정

	clnt_id	recency	frequency	monetary	R	F	M	RFMgroup	RFM_segment
0	2	30	6	157100	1	3	2	132	at risk
1	9	6	7	339941	3	3	3	333	need attention
2	12	38	1	29900	1	1	1	111	lost
3	20	1	7	238580	5	3	3	533	potential loyalists
4	23	3	9	202964	4	4	3	443	loyal customers

3.1 RFMgroup에 따른 RFM_segment 지정

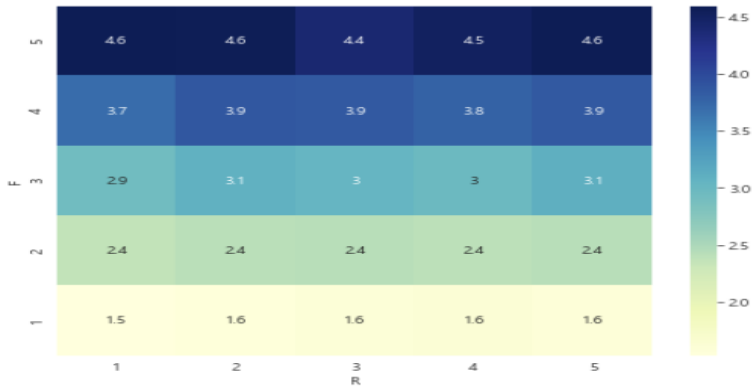
- recency, frquency, monetary 값을 5분위로 나누어 1~5점까지 점수를 부여 R, F, M 3개의 열을 만들고, 이 열들을 더하여 RFMgroup을 만들었음.
- RFMgroup을 바탕으로 옆의 R, F grid에 따라 RFM_segment를 지정하게 되는데, RFM_segment라는 새로운 열에 R과 F를 map해두고 정규식을 만들어 RFM_segment를 지정.



02

RFM 분석

3. RFM 분석



3.3 RFM_segment feature

- heatmap에서 알 수 있듯이 F가 높을수록 M이 우수해야하지만, 금액과 관련한 여러 feature들을 시각화해서 보면 알 수 있듯이 총구매액이 비교적 낮은 RFM_segment인 lost가 거래당구매액에서는 상대적으로 매우 높은 값을 가지는 것을 확인할 수 있음.
- 기업의 입장에서는 총구매액도 중요하지만 거래당구매액, 고가상품구매율 등 금액과 관련한 유의미한 값들이 더 많다고 생각.
- 금액과 관련한 유의미한 feature들을 통해 Ap_segment(amount price)를 만들어 한번 더 고객 세분화를 진행할 필요성이 있다고 생각함.

3. AP_segment 만들기

- AP_segment를 위한 features
- AP_segment 지정

03 AP_segment 만들기

1. AP_segment

5	DIAMOND
4	PLATINUM
3	GOLD
2	SLIVER
1	BRONZE

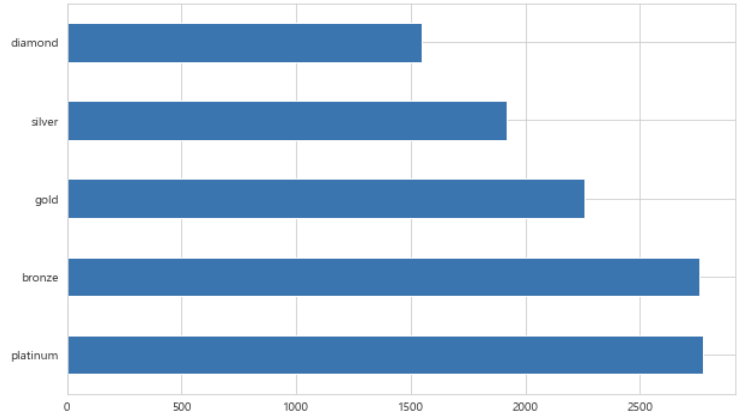
m grid

1.1 AP_segment 지정

- clnt_id별로 총구매액, 최대구매액, 최소구매액, 거래당구매액, 고가상품구매율, 저가상품구매율 feature를 만들었음.
- 각 해당 열별로 5분위수로 나누어 1~5점을 부여하고 점수를 합산하여 합산값을 바탕으로 다시 5분위수로 나누어 1~5점을 부여
- 이를 기준으로 아래의 m grid에 맞게 새로운 AP_segment를 지정함.
- 따라서 모든 고객은 각각 RFM_segment, AP_segment 총 2개의 segment를 가짐.

	clnt_id	RFM_segment	AP_segment
0	2	at risk	gold
1	9	need attention	platinum
2	12	lost	bronze
3	20	potential loyalists	silver
4	23	loyal customers	silver
...	...	...	...
11261	72373	loyal customers	diamond
11262	72400	hibernating	platinum
11263	72410	at risk	platinum
11264	72423	lost	platinum
11265	72424	need attention	platinum

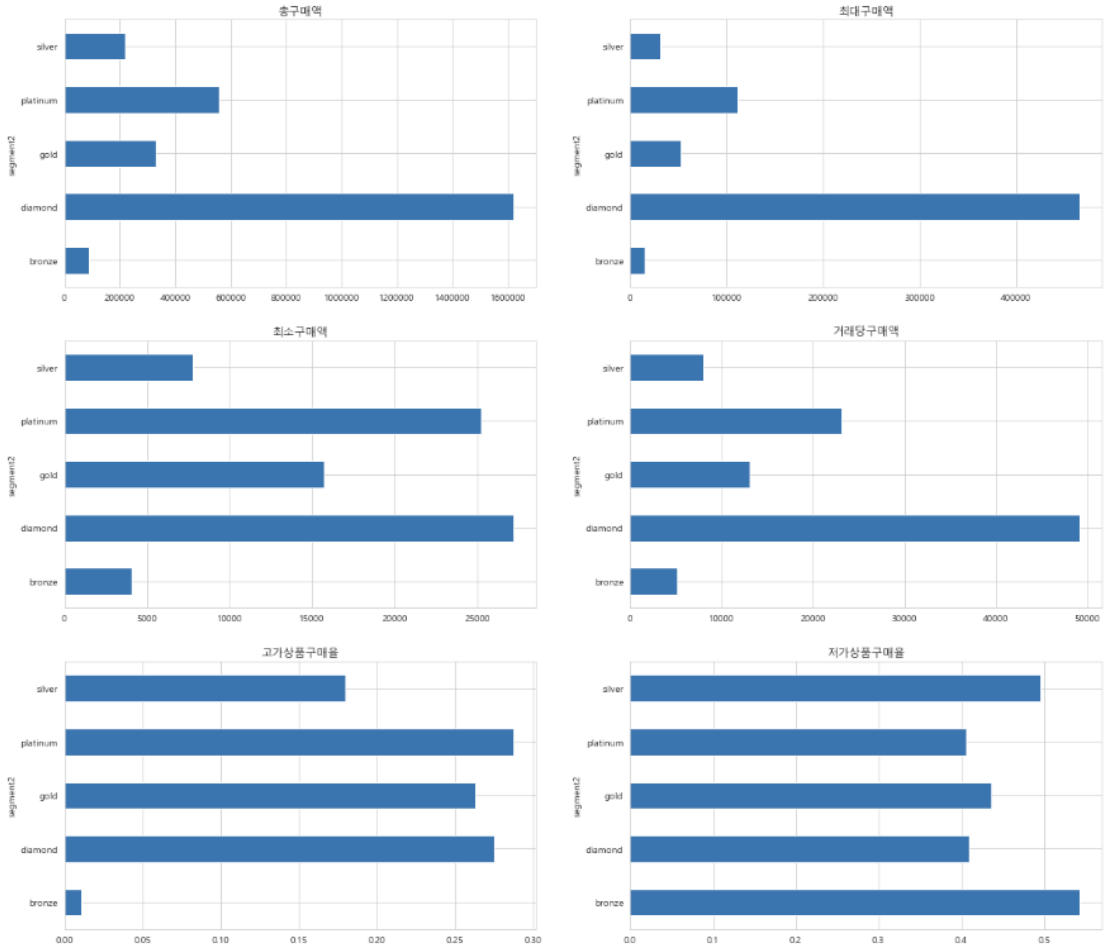
11266 rows × 3 columns



03

AP_segment 만들기

2. AP_segment feature



2.1 AP_segment별 고객 특성

- 아까 RFM_segment에서 봤던 feature의 결과를 다시 보면,
  이들이 AP_segment를 나누는 기준이 된 feature들이기때문에
  AP_segment가 우수할수록 총구매액, 거래당구매액 등의
  각 feature 순위가 높음을 확인할 수 있음.
- RFM_segment와 AP_segment를 통해 각 고객을
  RFM_segment와 AP_segment가 둘 다 우수한 **핵심 우량 고객군,**
  R과 AP_segment가 우수한**우량 가망 신규 고객군,**
  R을 제외한 값들이 우수한 **이탈위험 고객군,**
  RFM_segment와 AP_segment가 둘 다 낮은 **저수익성 고객군**으로
  나누어 볼 수 있는 고객 세분화가 이루어짐.

4. 고객별 상품추천

- segment별 상품추천
- 유사집단별 상품추천

04

고객별 상품추천

1. 고객별 상품추천 - segment별 상품추천

1.1 segment별 best seller 상품 추천

- segment가 같은 사람들을 유사집단으로 간주하고
  각 segment에 따라 상품을 추천함.
- RFM_segment와 AP_segment별 clac_nm3(상품소분류) 베스트셀러 top5를 뽑고
  함수를 만들어 각 고객에게 상품을 추천.
- 이때 고객이 기존에 구매했던 상품은 제거해주고
  RFM_segment_best와 AP_segment_best에서 중복된 상품도 제거하여
  segment_recommend_items를 만들었음.

	clnt_id	RFM_segment	AP_segment	RFM_segment_best	AP_segment_best
0	2	at risk	gold	[General Snacks, Ramens, Fresh Milk, Chicken E...	[General Snacks, Fresh Milk, Tofu, Ramens, Chi...
1	9	need attention	platinum	[General Snacks, Ramens, Chicken Eggs, Fresh M...	[General Snacks, Fresh Milk, Trash Bags, Chick...
2	12	lost	bronze	[General Snacks, Ramens, Chicken Eggs, Bibim R...	[General Snacks, Ramens, Chicken Eggs, Fresh M...
3	20	potential loyalists	silver	[General Snacks, Ramens, Chicken Eggs, Fresh M...	[General Snacks, Ramens, Chicken Eggs, Fresh M...
4	23	loyal customers	silver	[General Snacks, Fresh Milk, Chicken Eggs, Tof...	[General Snacks, Ramens, Chicken Eggs, Fresh M...
...	...	...	...	...	...
11261	72373	loyal customers	diamond	[General Snacks, Fresh Milk, Chicken Eggs, Tof...	[General Snacks, Trash Bags, Fresh Milk, Chick...
11262	72400	hibernating	platinum	[Ramens, General Snacks, Chicken Eggs, Corn Sn...	[General Snacks, Fresh Milk, Trash Bags, Chick...
11263	72410	at risk	platinum	[General Snacks, Ramens, Fresh Milk, Chicken E...	[General Snacks, Fresh Milk, Trash Bags, Chick...
11264	72423	lost	platinum	[General Snacks, Ramens, Chicken Eggs, Bibim R...	[General Snacks, Fresh Milk, Trash Bags, Chick...
11265	72424	need attention	platinum	[General Snacks, Ramens, Chicken Eggs, Fresh M...	[General Snacks, Fresh Milk, Trash Bags, Chick...

동일한 segment를 가진 사람들을 유사집단으로 보고
상품을 추천했지만,
고객이 구매했던 상품을 기준으로 유사집단을 나눠
추천한 것이 아니기때문에 구매한 상품이 비슷한 사람끼리
유사집단으로 보고 상품 추천을 한다면
더욱 효과적일 것이라고 생각하여 상품 추천을 한번 더 진행함.

04

고객별 상품추천

2. 고객별 상품추천 - 유사집단별 상품추천

```
# 고객별로 clac_nm3 구매 여부에 대한 pivot table (구매했으면 1, 안했으면 0 )
ratings_matrix = pd.pivot_table(df, index='clnt_id', columns='clac_nm2', values='pd_c',
                                aggfunc=lambda x: 1 if len(x) >= 1 else 0, fill_value=0)

print(ratings_matrix.shape)
ratings_matrix
```

```
...
```

```
# 유사도가 가장 높은 이웃의 수를 50으로 지정.
K = 50
```

```
# cosine_similarity()는 행을 기준으로 고객간 유사도를 구함.
user_sim = cosine_similarity(ratings_matrix, ratings_matrix)

# 상품명 매핑을 위한 행과 열이 동일한 데이터프레임이 생성됨.
user_sim = pd.DataFrame(user_sim, ratings_matrix.index, ratings_matrix.index)

# 고객이 구매했던 상품을 뽑아내기 위해 의도적으로 대각선 값을 1에서 2(코사인 유사도 최대값 1보다 크게)로 변경
np.fill_diagonal(user_sim.values, 2)

print(user_sim.shape)
user_sim
```

```
...
```

```
# 각 고객마다 K-nearest neighbors 생성. 이때 자기 자신을 가장 가까운 이웃으로 설정
# 행은 고객, 열은 K개의 이웃인 데이터 프레임 knn
knn = user_sim.apply(lambda x,k: x.sort_values(ascending=False).index[:k+1], args=(K,)).T
knn
```

```
...
```

```
# clac_nm2 즉, 상품중분류이기때문에 추천할 상품수는 5개로 지정.
N = 5
```

```
# 이미 구매한 상품을 제외하고 유사집단에서 가장 많이 구매한 N개의 상품을 추천
def top_n(x, n):
    purchased = purchased_list.filter(items=[x[0]]).iloc[0]  # 고객이 구매했던 상품 뽑기
    candidate = ratings_matrix.filter(items=x[1:], axis=0).sum().sort_values(ascending=False).index.to_list() # 이웃들이 가장 많이
    return [item for item in candidate if item not in purchased][:n]  # 이미 구매한 제품 제외하고 n개 추천

# 위의 함수를 knn에 적용하여 고객별 유사집단 추천상품열이 있는 데이터프레임 만들기
recommend_list = knn.apply(top_n, args=(N,), axis=1).reset_index().rename(columns={0:'recommend_items'})
recommend_list
```

2.1 유사집단별 추천상품

- 고객별로 clac_nm2(상품중분류) 구매 여부에 대한 pivot_table을 만들어 구매한 경우 1, 구매하지 않은 경우 0으로 값을 채움.
- cosine_similarity()를 이용하여 고객간 유사도를 구하고 각 고객마다 본인을 가장 가까운 이웃으로 하는 K-nearest neighbors 51명을 뽑아 knn 데이터 프레임을 만듦.
- def함수를 만들어 각 고객별로 이미 구매한 상품을 제외하고 유사집단에서 가장 많이 구매한 5개의 상품을 추천함.
- 구매 상품이 유사한 집단에서 상품을 추천한 것이기 때문에 매우 유용한 추천 방식이라고 생각함.

	clnt_id	recommend_items
0	2	[Frozen Instant Foods, Eggs, Chilled Instant F...
1	9	[Eggs, Chilled Instant Foods, Leaf Vegetables,...
2	12	[Snacks, Frozen Instant Foods, Skin Care, Writ...
3	20	[Tofu / Bean Sprouts, Domestic Fruits, Chilled...
4	23	[Instant Noodles, Snacks, Biscuits, Frozen Ins...
...	...	...
11261	72373	[Eggs, Yogurt, Frozen Instant Foods, Domestic ...
11262	72400	[Body Care, Women's Upper Bodywear / Tops, Jew...
11263	72410	[Instant Noodles, Frozen Instant Foods, Domest...
11264	72423	[Women's Upper Bodywear / Tops, Women's Underw...
11265	72424	[Domestic Fruits, Milk, Ham and Sausages, Impo...

11266 rows × 2 columns

5. 고객별 서비스추천

- segment를 이용한 group
- group별 서비스 추천

05

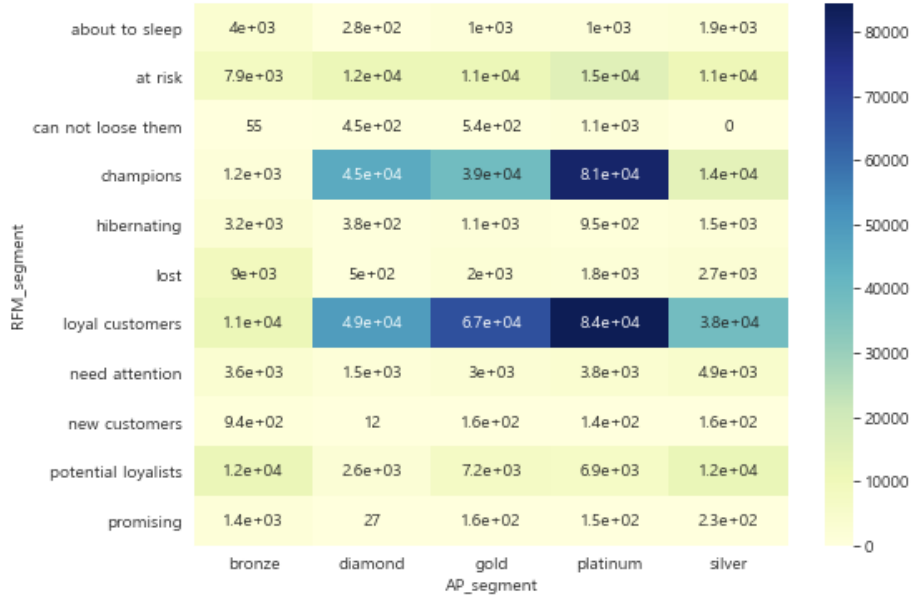
고객별 서비스추천

1. segment를 이용한 group화

1.1 segment를 이용한 group만들기

- RFM_segment와 AP_segment 사이의 상관관계를 보여주는 heatmap을 통해 알 수 있듯이 AP_segment는 RFM_segment와는 달리 금액만 고려한 segment이기 때문에 RFM_segment와 AP_segment 사이의 상관관계가 높은것이 거의 존재하지 않음.
- 따라서 임의로 RFM_segment와 AP_segment를 이용하여 group을 나눔.

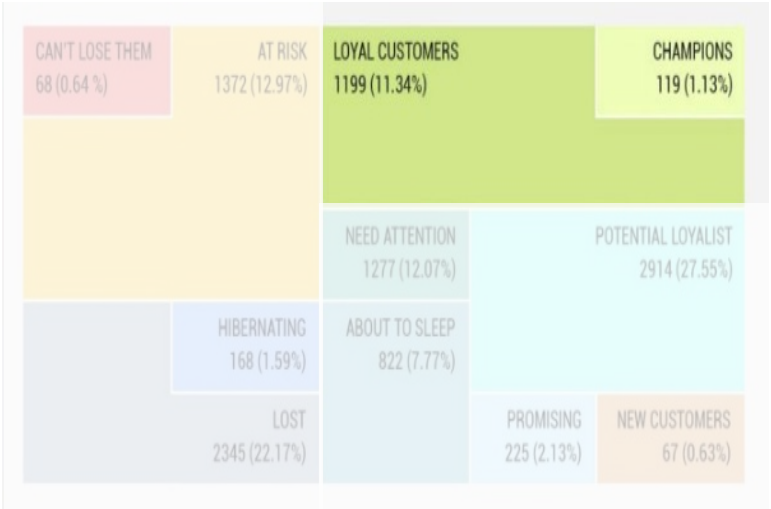
```
group1 = df2.query('segment in ["champions","loyal customers"] and segment2 in ["diamond", "platinum"]')
group2 = df2.query('segment in ["promising","new customers"] and segment2 in ["diamond", "platinum","gold", "sliver"]')
group3 = df2.query('segment in ["can not loose them","at risk"] and segment2 in ["diamond", "platinum","gold", "sliver"]')
group4 = df2.query('segment in ["lost", "hibernating", "about to sleep"] and segment2 in ["bronze"]')
```

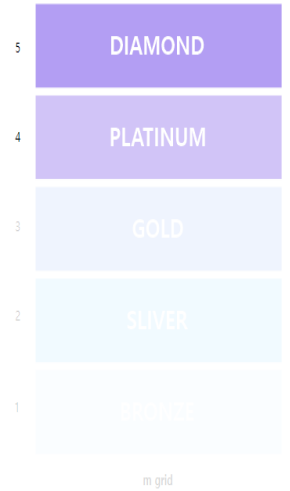


- group1은 RFM_segment와 AP_segment 모두 우수한 핵심 우량 고객군
- group2는 R이 우수하고 AP_segment도 우수한 편에 속하는 우량 가망 신규고객군
- group3는 AP_segment는 우수하나 RFM_segment는 낮은 이탈위험 고객군
- group4는 RFM_segment와 AP_segment 모두 낮은 저 수익성 고객군
- 나머지는 일반고객으로 봄.

05

고객별 서비스추천





2. group별 서비스 추천

2.1 핵심 우량 고객군 서비스 추천

- 핵심 우량 고객군의 경우,
  해당 고객들이 핵심 우량 고객군에서 이탈하지 않고 유지하도록
  프리미엄 서비스를 제공해야함.
- 고가 제품들에 대하여 구매 혜택(높은 적립금, 사은품) 제공
- 여러 프로모션의 우선순위 대상이 될 수 있는 vvip card 제공 등의 서비스가 필요.

```
h_price = group1.groupby('pd_c')['gd_price'].mean().quantile(0.999)
group1.query('gd_price >= @h_price').clac_nm1.unique()
```

```
array(['Canned / Jarred Foods', "Women's Clothing",
       'Heating / Cooling Electronics', 'Health Care',
       'Refrigerators and Washing Machines', 'Coffee / Tea',
       'Fashion Accessories', 'Video / Audio System Electronics'],
      dtype=object)
```

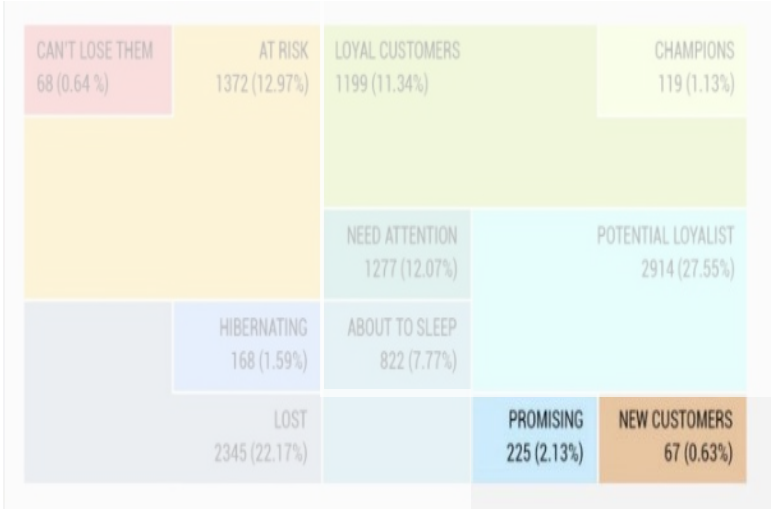
- 예를 들어, 상위 0.001%에 해당하는 고가의 제품을 구매한 경우,
  구매단가가 상위 55%~ 75%에 해당하는 제품들 중
  베스트셀러를 뽑아 고객이 사은품으로 선택할 수 있도록 하는 서비스.

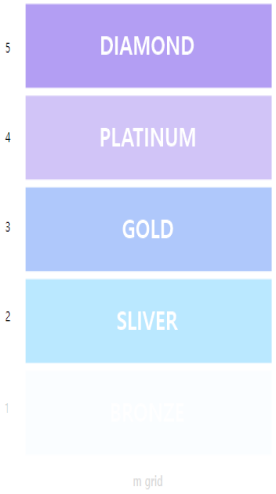
```
m1_price = group1.groupby('pd_c')['gd_price'].mean().quantile(0.55)
m2_price = group1.groupby('pd_c')['gd_price'].mean().quantile(0.75)
group1.query('@m1_price <= gd_price <= @m2_price').clac_nm3.value_counts().head(3).index
```

```
Index(['Frozen Dumplings', 'Domestic Porks - Bellys', 'Peaches'], dtype='object')
```

05

고객별 서비스추천

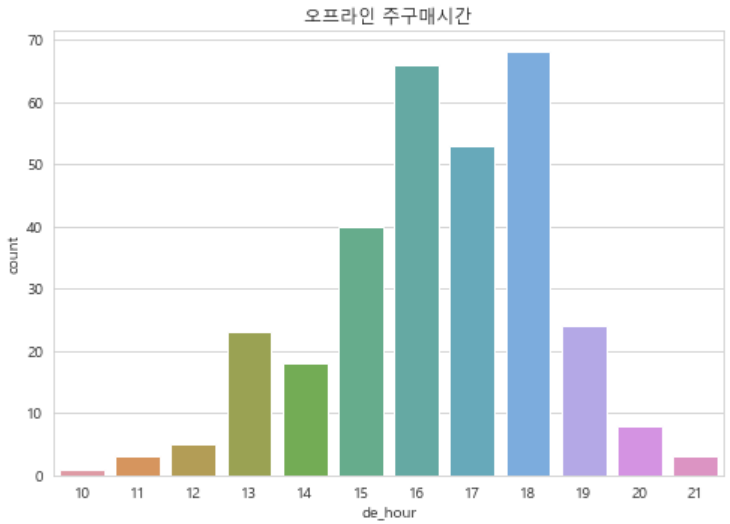


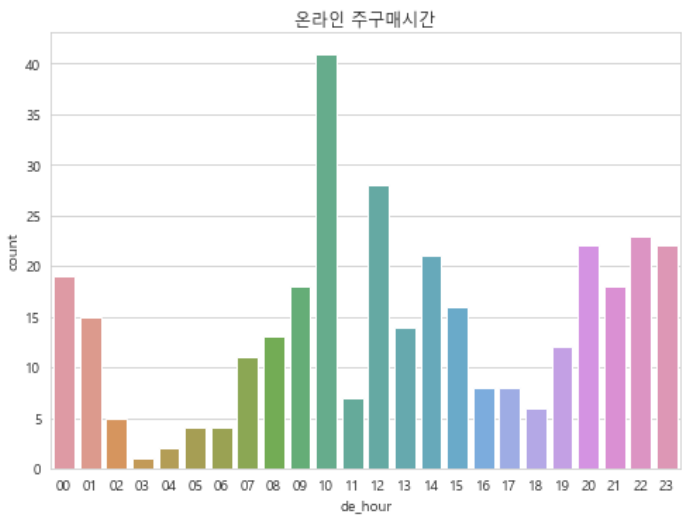


2. group별 서비스 추천

2.2 우량 가망 신규고객군 서비스 추천

- 우량 가망 신규고객군의 경우,
  구매빈도 즉, F를 높여 핵심 우량 고객군으로 성장시키기 위해
  구매를 촉진하여 구매 횟수를 늘리도록 하는 서비스들이 필요함.





- 예를 들어,
  온라인과 오프라인 주구매시간을 나눠
  오프라인으로는 오후 4시부터 오후 7시까지
  타임세일을 진행하고
  온라인은 오전 10시 1시간, 오후 8시부터 자정까지
  깜짝 할인 쿠폰을 제공함.

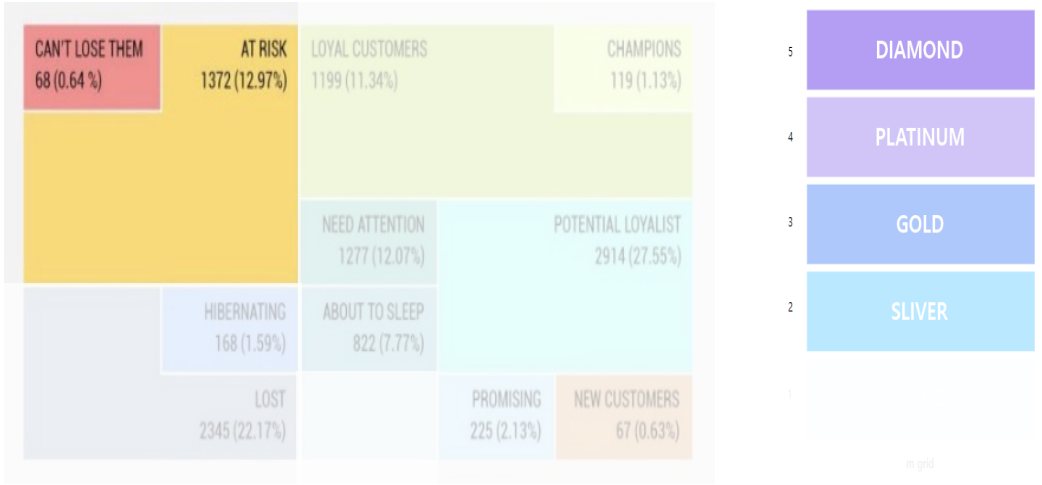
05

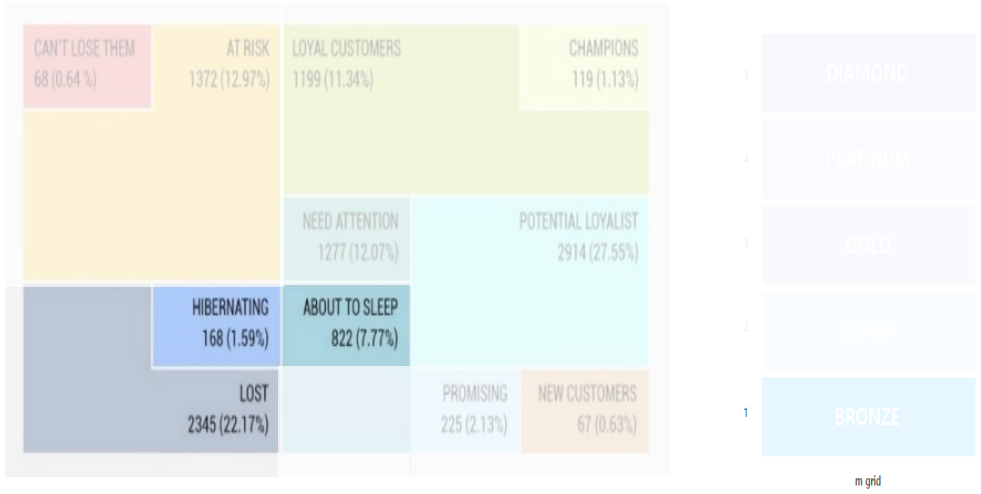
고객별 서비스추천

2. group별 서비스 추천

2.3 이탈위험 고객군 서비스 추천

- 이탈위험 고객군은 재활성화를 위하여
앞에서 언급했던 할인이나 사은품 프로모션을 진행함과 동시에
프로모션 진행사실을 SMS로 보내 홍보하여
고객으로 하여금 이 사실을 인지할 수 있도록 해야함.





2.4 저수익 고객군 서비스 추천

- 저수익 고객군에게는 맞춤형 서비스를 제공하지 않고
마케팅 비용을 절감하여
다른 고객군을 위한 서비스에 집중하는 것이 효율적임.

감사합니다.

분석프로그래밍 프로젝트 - 분석보고서

빅데이터경영통계전공 20192761 김정하


Q&A