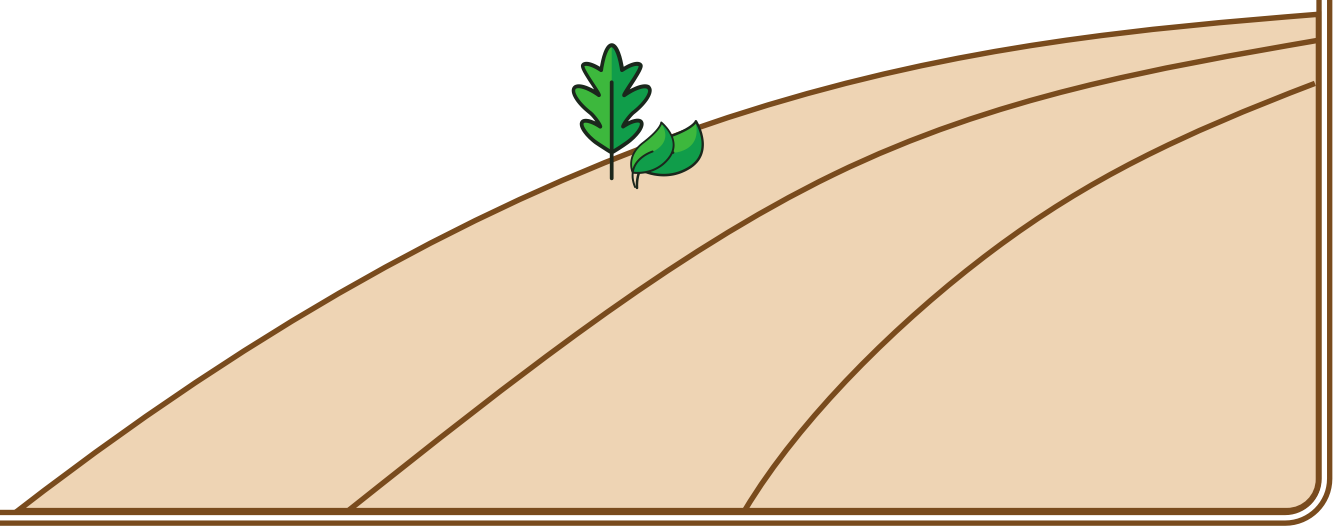
3월 BC카드 이용내역 데이터분석

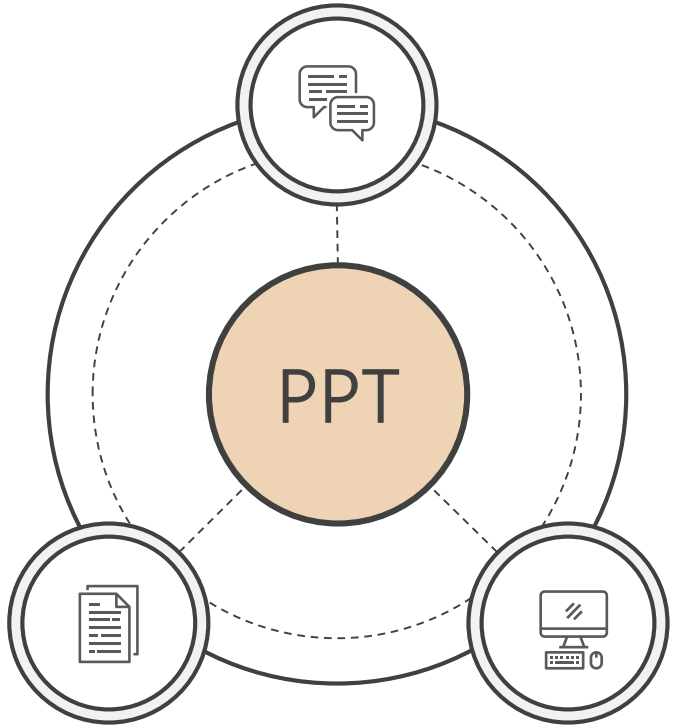
데이터분석 및 머신러닝을 통해 다양한 결과 도출하기



18102002 이현진

CONTENTS1

데이터 전처리 및 간단한 데이터 분석



CONTENTS2

시각화

CONTENTS3

머신러닝(DecisionTree)

CONTENTS1 데이터 전처리 및 간단한 데이터 분석

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	REG_YYMI	MEGA_CT	MEGA_CT	CTY_RGN_	CTY_RGN_	ADMI_CTY	ADMI_CTY	MAIN_BUZ	MAIN_BUZ	TP_GRP_N	TP_GRP_N	TP_BUZ_N	TP_BUZ_N	CSTMR_GU	CSTMR_M	CSTMR_M	CSTMR_CT	CSTM
2	202003	11	서울특별시	1168	강남구	11680750	수서동	30	생활	40	유통업영리	4010	편 의 점	내국인	11	서울특별시	1168	강남구
3	202003	11	서울특별시	1129	성북구	11290660	길음1동	30	생활	70	의료기관	7041	약국	내국인	11	서울특별시	1129	성북구
4	202003	11	서울특별시	1144	마포구	11440555	아현동	30	생활	40	유통업영리	4076	인터넷 P/(	내국인	11	서울특별시	1114	중구
5	202003	11	서울특별시	1126	중랑구	11260590	상봉2동	30	생활	40	유통업영리	4010	편 의 점	내국인	11	서울특별시	1126	중랑구
6	202003	11	서울특별시	1168	강남구	11680630	대치4동	30	생활	62	보험	6201	생명 보험	내국인	41	경기도	4146	용인시
7	202003	11	서울특별시	1130	강북구	11305660	인수동	30	생활	92	수리서비스	9210	세탁소	내국인	11	서울특별시	1130	강북구
8	202003	11	서울특별시	1117	용산구	11170650	이태원1동	80	음식	80	일반음식	8001	일반한식	내국인	11	서울특별시	1138	은평구
9	202003	11	서울특별시	1135	노원구	11350640	상계2동	80	음식	80	일반음식	8001	일반한식	내국인	11	서울특별시	1135	노원구
10	202003	11	서울특별시	1159	동작구	11590540	상도2동	80	음식	80	일반음식	8001	일반한식	내국인	11	서울특별시	1159	동작구
11	202003	11	서울특별시	1130	강북구	11305545	송중동	30	생활	70	의료기관	7020	의원	내국인	11	서울특별시	1129	성북구
12	202003	11	서울특별시	1117	용산구	11170625	한강로동	30	생활	40	유통업영리	4076	인터넷 P/(	내국인	41	경기도	4111	수원시
13	202003	11	서울특별시	1168	강남구	11680650	역삼2동	30	생활	40	유통업영리	4076	인터넷 P/(	내국인	11	서울특별시	1171	송파구
14	202003	11	서울특별시	1156	영등포구	11560540	여의동	80	음식	80	일반음식	8006	서양음식	내국인	11	서울특별시	1168	강남구
15	202003	11	서울특별시	1156	영등포구	11560540	여의동	50	내구재	31	가전제품	3101	가전 제품	내국인	41	경기도	4117	안양시
16	202003	11	서울특별시	1156	영등포구	11560540	여의동	50	내구재	31	가전제품	3101	가전 제품	내국인	11	서울특별시	1121	광진구

사용한 데이터 : bc_card_202003_out.txt

CONTENTS1 데이터 전처리 및 간단한 데이터 분석

```
import pandas as pd
import csv
import numpy as np
```

```
bc = pd.read_csv('bccard_202003.csv',encoding = 'cp949')
bc.columns=['년월','이용시도_코드','이용시도',
                  '이용시군구_코드','이용시군구','이용읍면동_코드',
                  '이용읍면동','업종대분류_코드','업종대분류',
                  '업종중분류_코드','업종중분류','업종소분류_코드','업종소분류',
                  '내외국인','고객거주시도_코드','고객거주시도',
                   '고객거주시군구_코드','고객거주시군구','성별','연령대',
                   '가구생애주기','이용금액','이용건수']
```

데이터 칼럼 이름이 어떤 의미인지 알아보기 어려워

데이터명세서를 참고하여 칼럼 이름을 다시 설정

CONTENTS1 데이터 전처리 및 간단한 데이터 분석

	년월	이용시도_코드	이용시도	이용시군구_코드	이용시군구	이용읍면동_코드	이용읍면동	업종대분류_코드	업종대분류	업종중분류_코드	...	내외국인	고객거주시도_코드	고객거주시도	고객거주시군구_코드	고객거주시군구	성별	연령대	가구생애주기	이용금액	이용건수
0	202003	11.0	서울특별시	1168.0	강남구	11680750.0	수서동	30.0	생활	40.0	...	내국인	11.0	서울특별시	1168.0	강남구	2.0	20대	1.0	7927440.0	1089
1	202003	11.0	서울특별	1129.0	성북구	11290660.0	길음1동	30.0	생활	70.0	...	내국인	11.0	서울특별	1129.0	성북구	1.0	40대	4.0	274100.0	25

CONTENTS1 데이터 전처리 및 간단한 데이터 분석

```
# 모든 행이 동일한 데이터값을 가지고 있는 열 삭제하기
bc.drop(['년월','이용시도_코드','이용시도','내외국인'],axis=1,inplace=True)
```

```
bc
```

	이용시군구_코드	이용시군구	이용읍면동_코드	이용읍면동	업종대분류_코드	업종대분류	업종중분류_코드	업종중분류	업종소분류_코드	업종소분류	고객거주시도_코드	고객거주시도	고객거주시군구_코드	고객거주시군구	성별	연령대	가구생애주기	이용금액	이용건수
0	1168.0	강남구	11680750.0	수서동	30.0	생활	40.0	유통업영리	4010.0	편의점	11.0	서울특별시	1168.0	강남구	2.0	20대	1.0	7927440.0	1089
1	1129.0	성북구	11290660.0	길음1	30.0	생활	70.0	의료기	7041.0	약국	11.0	서울특별	1129.0	성북구	1.0	40대	4.0	274100.0	25

CONTENTS1 데이터 전처리 및 간단한 데이터 분석

```
# 지역 코드를 보면 한눈에 보이지 않기 때문에 파악을 쉽게 하기 위해 코드 열을 없앤다.
bc03=bc.drop(['이용시군구_코드','이용읍면동_코드','업종대분류_코드','업종중분류_코드',
              '업종소분류_코드','고객거주시도_코드','고객거주시군구_코드'],axis=1)
bc03
```

	이용시군구	이용읍면동	업종대분류	업종중분류	업종소분류	고객거주시도	고객거주시군구	성별	연령대	가구생애주기	이용금액	이용건수
0	강남구	수서동	생활	유통업영리	편의점	서울특별시	강남구	2.0	20대	1.0	7927440.0	1089
1	성북구	길음1동	생활	의료기관	약국	서울특별시	성북구	1.0	40대	4.0	274100.0	25
2	마포구	아현동	생활	유통업영리	인터넷P/G	서울특별시	중구	1.0	60대 이상	5.0	34395725.0	808
3	중랑구	상봉2동	생활	유통업영리	편의점	서울특별시	중랑구	2.0	20대	1.0	31860800.0	3699
4	강남구	대치4동	생활	보험	생명보험	경기도	용인시	2.0	50대	4.0	2546487.0	45
...	...	...	...	...	...	...	...	...	...	...	...	...

가정

* 카드이용매출액이 가장 낮은 지역을 고르고 그 지역의 매출액을 높이기 위한 이벤트를 개발하려고 한다.
* 특정 지역에 한해서 일정금액 이상을 결제할 경우 일부 할인해주는 이벤트를 하려고 할 때 어느 지역에 해당 이벤트를 적용하는 것이 좋을까?

```
이용금액 = bc03.groupby('이용시군구')['이용금액'].sum()
print(min(이용금액.index),min(이용금액))
```

```
강남구 7106680030.0
```

CONTENTS2 시각화

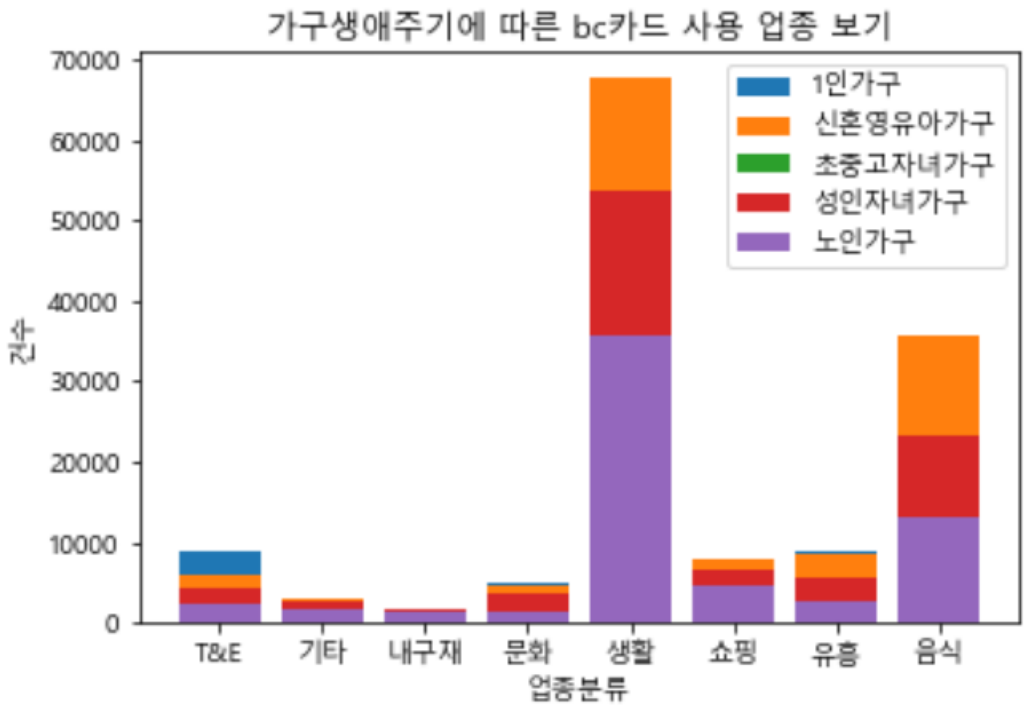
```
df =pd.DataFrame(bc03.groupby(['가구생애주기','업종대분류']).count().iloc[:,0])
df.columns=['건수']
가구생애주기 = []
업종대분류=[]
for i in range(len(list(df.index))):
    가구생애주기.append(list(df.index)[i][0])
    업종대분류.append(list(df.index)[i][1])
df.index=list(range(0,40))
df['가구생애주기']=가구생애주기
df['업종대분류']=업종대분류
df
```

	건수	가구생애주기	업종대분류
0	8835	1.0	T&E
1	2678	1.0	기타
2	1070	1.0	내구재
3	4898	1.0	문화
4	53557	1.0	생활

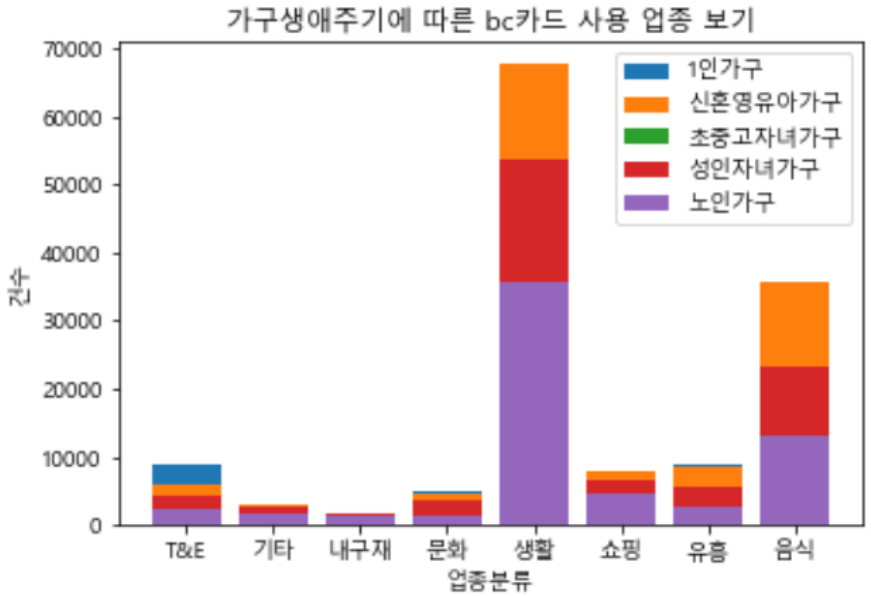
가구 생애 주기에 따라 bc카드 사용 업종에 차이가 있을까?

CONTENTS2 시각화

```
df1 = df.loc[df['가구생애주기']==1,['건수','업종대분류']]
df2 = df.loc[df['가구생애주기']==2,['건수','업종대분류']]
df3 = df.loc[df['가구생애주기']==3,['건수','업종대분류']]
df4 = df.loc[df['가구생애주기']==4,['건수','업종대분류']]
df5 = df.loc[df['가구생애주기']==5,['건수','업종대분류']]

import matplotlib.pyplot as plt

p1 = plt.bar(df1['업종대분류'],df1['건수'],label='1인가구')
p2 = plt.bar(df2['업종대분류'],df2['건수'],label='신혼영유아가구')
p3 = plt.bar(df3['업종대분류'],df3['건수'],label='초중고자녀가구')
p4 = plt.bar(df4['업종대분류'],df4['건수'],label='성인자녀가구')
p5 = plt.bar(df5['업종대분류'],df5['건수'],label='노인가구')
plt.title("가구생애주기에 따른 bc카드 사용 업종 보기")
plt.xlabel('업종분류')
plt.ylabel('건수')
plt.legend()
plt.show()
```



CONTENTS2 시각화



* 노인가구 - 생활에 필요한 생활용품이나 음식 업종에 비중이 크다.
* 전반적으로도 여유가 조금 더 있는 노인가구의 소비가 각 분야에서 많은 비중을 차지
* 신혼영유아가구 역시 새 살림을 하고 아이를 키우기 때문에 생활에 필요한 업종에 비중이 크다.
* 초중고자녀가구가 문화에 많은 비중을 차지할 것이라는 것과 1인가구가 쇼핑을 많이 차지할 것이라는 예상은 빗나갔다.

CONTENTS2 시각화

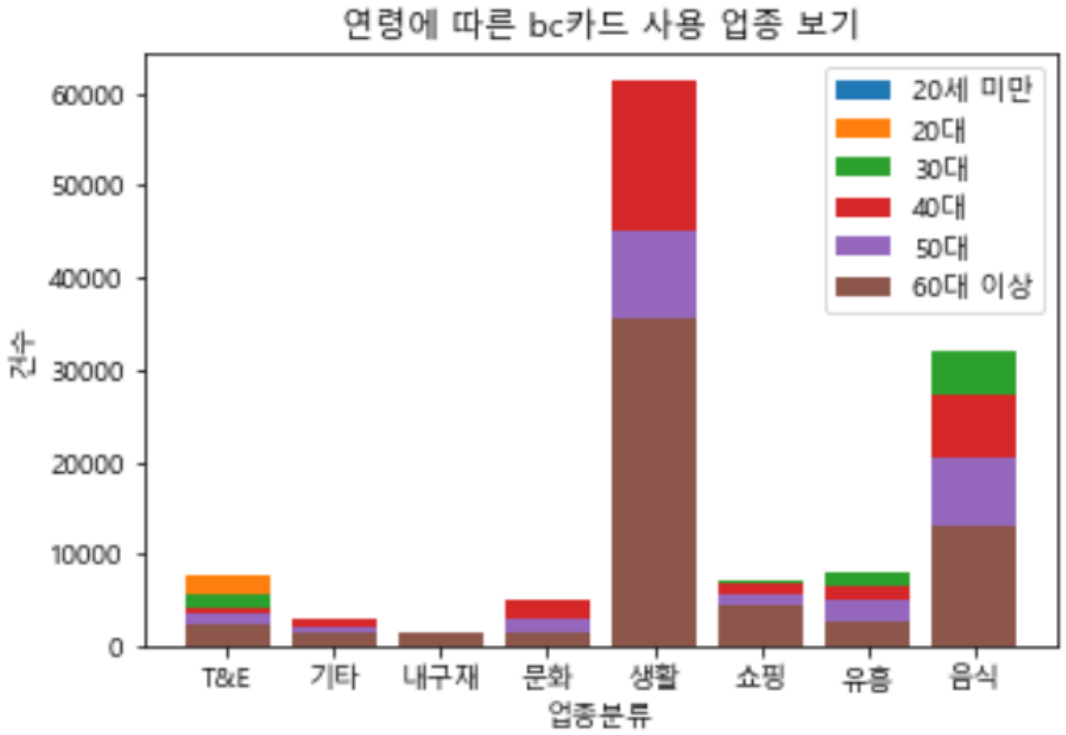
```
df =pd.DataFrame(bc03.groupby(['업종대분류','연령대']).count().iloc[:,0])
df.columns=['건수']
연령대 = []
업종대분류=[]
for i in range(len(list(df.index))):
    업종대분류.append(list(df.index)[i][0])
    연령대.append(list(df.index)[i][1])
df.index=list(range(0,48))
df['연령대']=연령대
df['업종대분류']=업종대분류
df
```

	건수	연령대	업종대분류
0	7623	20대	T&E
1	932	20세 미만	T&E
2	5521	30대	T&E
3	4299	40대	T&E

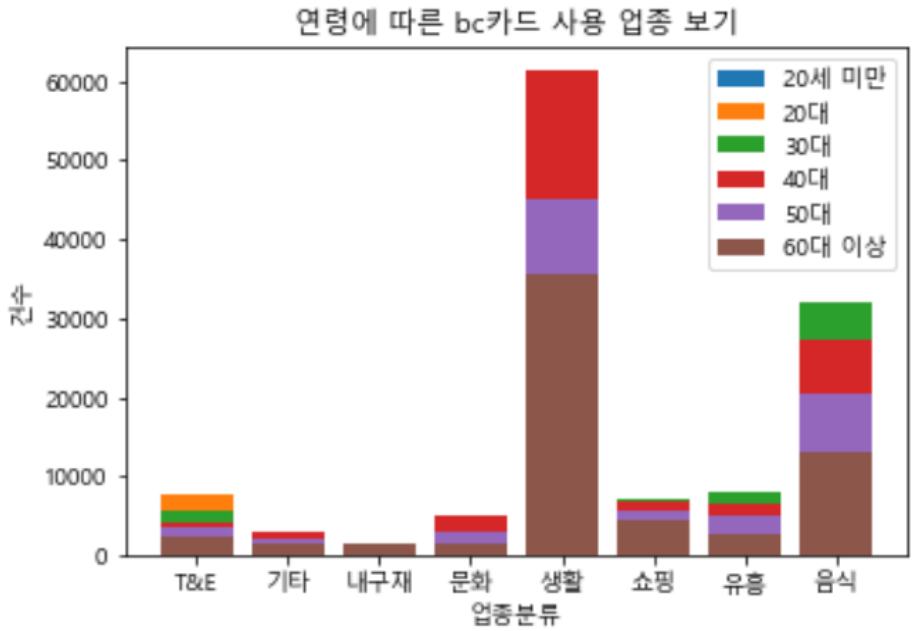
가구 생애 주기와 비슷한 맥락으로,
연령대에 따른 bc카드 이용 업종분류를
살펴보면 비슷한 결과가 나올까?

CONTENTS2 시각화

```
df1 = df.loc[df['연령대']=='20세 미만',['건수','업종대분류']]
df2 = df.loc[df['연령대']=='20대',['건수','업종대분류']]
df3 = df.loc[df['연령대']=='30대',['건수','업종대분류']]
df4 = df.loc[df['연령대']=='40대',['건수','업종대분류']]
df5 = df.loc[df['연령대']=='50대',['건수','업종대분류']]
df6 = df.loc[df['연령대']=='60대 이상',['건수','업종대분류']]

import matplotlib.pyplot as plt

p1 = plt.bar(df1['업종대분류'],df1['건수'],label='20세 미만')
p2 = plt.bar(df2['업종대분류'],df2['건수'],label='20대')
p3 = plt.bar(df3['업종대분류'],df3['건수'],label='30대')
p4 = plt.bar(df4['업종대분류'],df4['건수'],label='40대')
p5 = plt.bar(df5['업종대분류'],df5['건수'],label='50대')
p6 = plt.bar(df6['업종대분류'],df6['건수'],label='60대 이상')
plt.title("연령에 따른 bc카드 사용 업종 보기")
plt.xlabel('업종분류')
plt.ylabel('건수')
plt.legend()
plt.show()
```

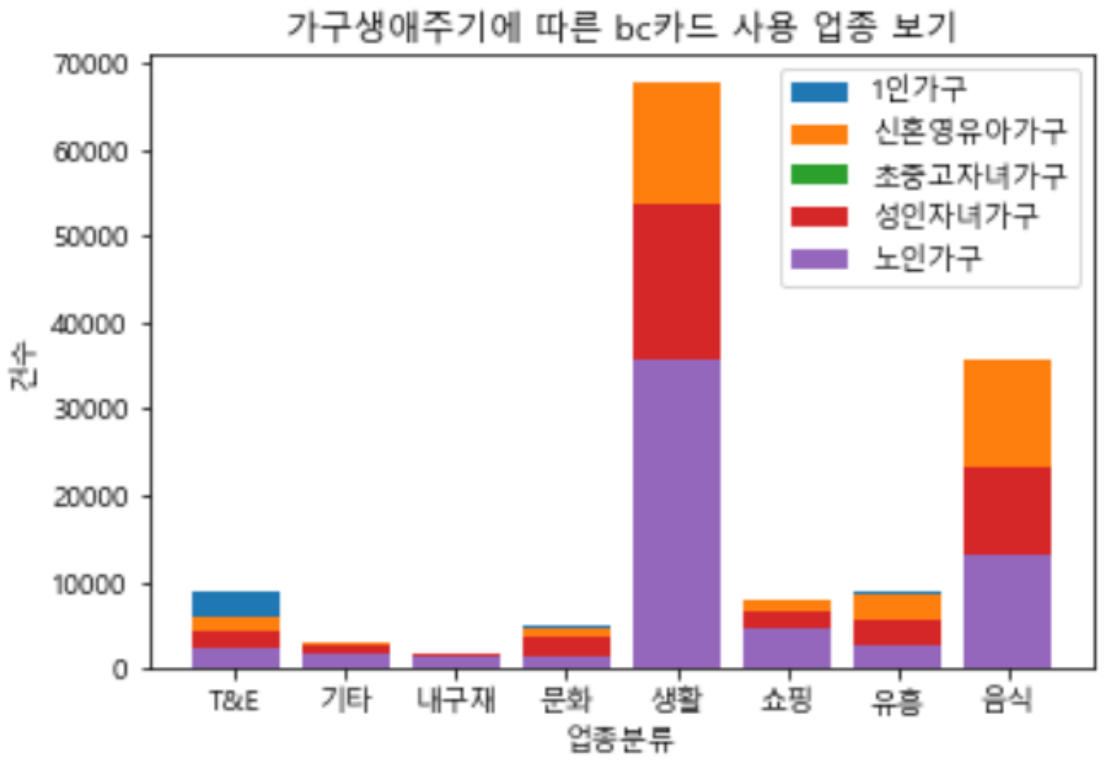


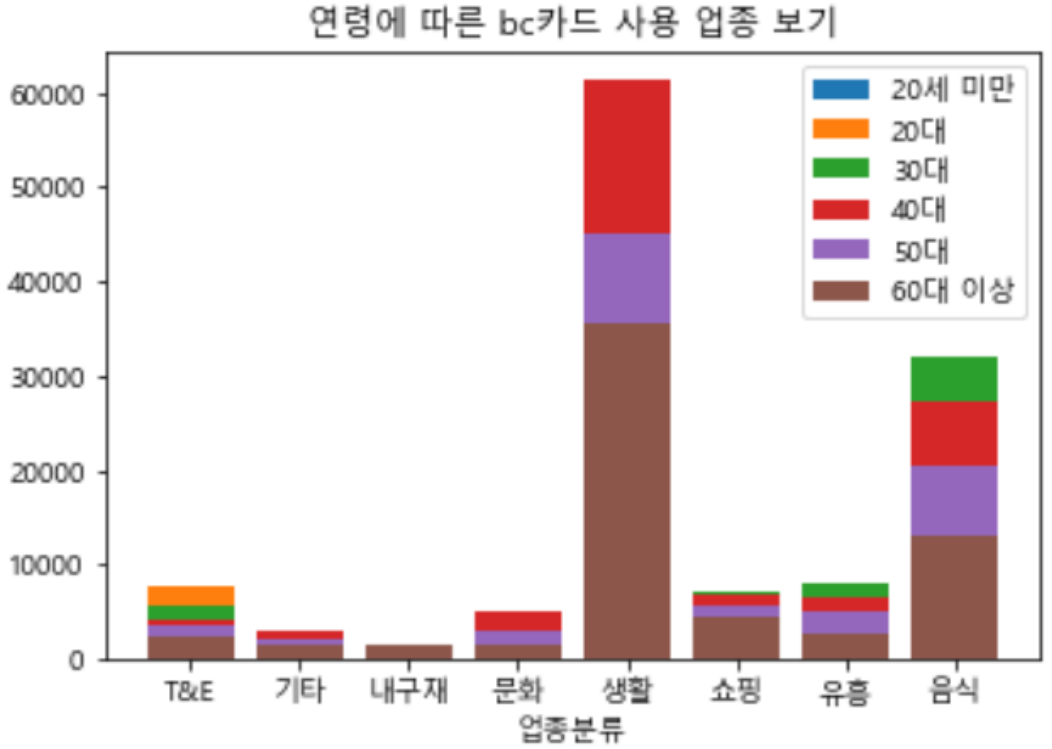
CONTENTS2 시각화



* 60대 이상 - 생활에 필요한 생활용품이나 음식 업종에 비중이 크다.
* 전반적으로도 여유가 조금 더 있는 60대 이상의 소비가 각 분야에서 많은 비중을 차지
* 20대가 문화에 많은 비중을 차지할 것이라는 예상은 빗나갔다.

CONTENTS2 시각화





CONTENTS2 시각화

```
df =pd.DataFrame(bc03.groupby(['업종대분류','성별']).count().iloc[:,0])
df.columns=['건수']
df1 =pd.DataFrame(bc03.groupby(['업종대분류']).count().iloc[:,0])
df1.columns=['건수']
df
```

		건수
업종대분류	성별	
T&E	1.0	15672
	2.0	8632
기타	1.0	6549
	2.0	5415
내구재	1.0	3792
	2.0	2926

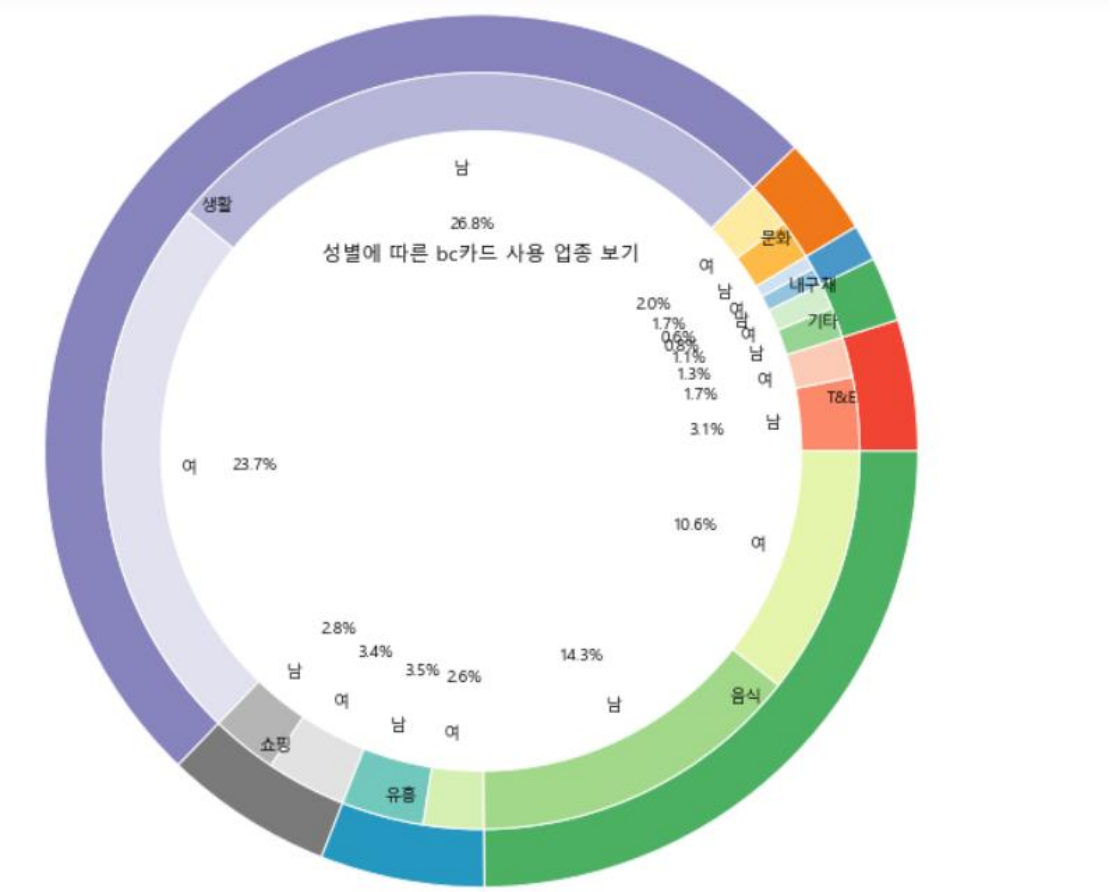
bc카드 사용 업종에 성별에 따른 차이가 존재할까?

CONTENTS2 시각화

```
group_names=['T&E','기타','내구재','문화','생활','쇼핑','유흥','음식']
group_sizes=[24304,11964,6718,18290,252674,31216,30421,124413]
subgroup_names=['남','여',
                '남','여',
                '남','여',
                '남','여',
                '남','여',
                '남','여',
                '남','여',
                '남','여']
subgroup_sizes=[15672,8632,6549,5415,3792,2926,8468,9822,134230,118444,14141,17075,17528,12893,71287,53126]
a, b, c, d, e, f, g, h = [
    plt.cm.Reds, plt.cm.Greens, plt.cm.Blues,
    plt.cm.YlOrBr, plt.cm.Purples, plt.cm.Greys,
    plt.cm.YlGnBu, plt.cm.YlGn]
width_num = 0.4

fig, ax = plt.subplots()
ax.axis('equal')
pie_outside, _ = ax.pie(group_sizes,
                        radius=3.0,
                        labels=group_names,
                        labeldistance=0.8,
                        colors=[a(0.6), b(0.6), c(0.6),
                                d(0.6), e(0.6), f(0.6),
                                g(0.6), h(0.6)])

plt.setp(pie_outside,
         width=width_num,
         edgecolor='white')
```

```
# Inside Ring
pie_inside, plt_labels, junk = ₩
    ax.pie(subgroup_sizes,
           radius=(3.0 - width_num),
           labels=subgroup_names,
           labeldistance=0.75,
           autopct='%1.1f%%',
           colors=[a(0.4), a(0.2),
                   b(0.4), b(0.2),
                   c(0.4), c(0.2),
                   d(0.4), d(0.2),
                   e(0.4), e(0.2),
                   f(0.4), f(0.2),
                   g(0.4), g(0.2),
                   h(0.4), h(0.2),
                   ])

plt.setp(pie_inside,
         width=width_num,
         edgecolor='white')
plt.title('성별에 따른 bc카드 사용 업종 보기')
plt.show()
```

CONTENTS2 시각화

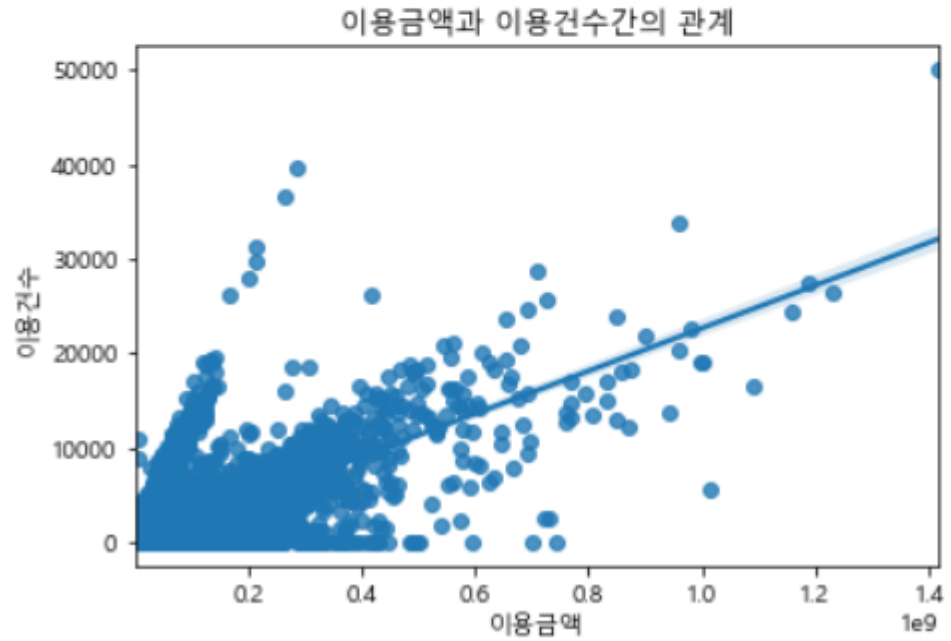


성별에 따른 차이가 크게 존재하지는 않다고 생각된다.

CONTENTS2 시각화

```
import seaborn as sns

ax=sns.regplot(x='이용금액',y='이용건수',data=bc03)
ax.set_xlabel('이용금액')
ax.set_ylabel('이용건수')
ax.set_title('이용금액과 이용건수간의 관계')
```

```
Text(0.5, 1.0, '이용금액과 이용건수간의 관계')
```



이용건수가 많을수록 이용금액이 많을 것이라고 가정.

그래프로 나타낸 결과 대체로 그러한 경향성은 보이지만 꼭 비례한다고 보기에는 다소 퍼짐의 정도가 크다.

CONTENTS3 머신러닝(DecisionTree)

<카드사 입장에서는 이용금액이 클수록 더 좋다.>
어떤 특징의 사람들이 카드 이용금액이 높은지 판단해보기. By DecisionTree

```
bc3=bc.drop(['이용시군구','이용읍면동','업종대분류','업종중분류','업종소분류','고객거주시도','고객거주시군구'],axis=1)
bc3
```

	이용시군구_코드	이용읍면동_코드	업종대분류_코드	업종중분류_코드	업종소분류_코드	고객거주시도_코드	고객거주시군구_코드	성별	연령대	가구생애주기	이용금액	이용건수
0	1168.0	11680750.0	30.0	40.0	4010.0	11.0	1168.0	2.0	20대	1.0	7927440.0	1089
1	1129.0	11290660.0	30.0	70.0	7041.0	11.0	1129.0	1.0	40대	4.0	274100.0	25
2	1144.0	11440555.0	30.0	40.0	4076.0	11.0	1114.0	1.0	60대 이상	5.0	34395725.0	808
3	1126.0	11260590.0	30.0	40.0	4010.0	11.0	1126.0	2.0	20대	1.0	31860800.0	3699
4	1168.0	11680630.0	30.0	62.0	6201.0	41.0	4146.0	2.0	50대	4.0	2546487.0	45
...	...	...	...	...	...	...	...	...	...	...	...	...
499995	1165.0	11650580.0	30.0	40.0	4010.0	11.0	1153.0	1.0	50대	4.0	38550.0	11
499996	1123.0	11230536.0	80.0	80.0	8005.0	11.0	1120.0	2.0	40대	2.0	189000.0	9
499997	1168.0	11680750.0	30.0	40.0	4076.0	43.0	4313.0	1.0	40대	4.0	601000.0	25
499998	1117.0	11170690.0	30.0	61.0	6140.0	11.0	1150.0	1.0	40대	3.0	9200.0	4

CONTENTS3 머신러닝(DecisionTree)

DecisionTreeClassifier를 사용하기 위해서는 문자열 데이터를 바꾸어주어야 한다.

```
bc3['연령대']=bc3['연령대'].replace({'20세 미만':10,'20대':20,'30대':30,'40대':40,'50대':50,'60대 이상':60})
bc3
```

	이용시군구_코드	이용읍면동_코드	업종대분류_코드	업종중분류_코드	업종소분류_코드	고객거주시도_코드	고객거주시군구_코드	성별	연령대	가구생애주기	이용금액	이용건수
0	1168.0	11680750.0	30.0	40.0	4010.0	11.0	1168.0	2.0	20	1.0	7927440.0	1089
1	1129.0	11290660.0	30.0	70.0	7041.0	11.0	1129.0	1.0	40	4.0	274100.0	25
2	1144.0	11440555.0	30.0	40.0	4076.0	11.0	1114.0	1.0	60	5.0	34395725.0	808
3	1126.0	11260590.0	30.0	40.0	4010.0	11.0	1126.0	2.0	20	1.(		
4	1168.0	11680630.0	30.0	62.0	6201.0	41.0	4146.0	2.0	50	4.(		
...	...	...	...	...	...	...	...	...	...	..		
499995	1165.0	11650580.0	30.0	40.0	4010.0	11.0	1153.0	1.0	50	4.(		
499996	1123.0	11230536.0	80.0	80.0	8005.0	11.0	1120.0	2.0	40	2.(		
499997	1168.0	11680750.0	30.0	40.0	4076.0	43.0	4313.0	1.0	40	4.(		
499998	1117.0	11170690.0	30.0	61.0	6140.0	11.0	1150.0	1.0	40	3.(		
499999	1111.0	11110530.0	30.0	61.0	6140.0	11.0	1171.0	1.0	40	2.(		

```
bc3.loc[:,:] = bc3.loc[:,:].astype(int)
```

```
bc3.dtypes
```

```
이용시군구_코드       int32
이용읍면동_코드       int32
업종대분류_코드       int32
업종중분류_코드       int32
업종소분류_코드       int32
고객거주시도_코드      int32
고객거주시군구_코드     int32
성별              int32
연령대             int32
가구생애주기          int32
이용금액            int32
이용건수            int32
dtype: object
```

CONTENTS3 머신러닝(DecisionTree)

이용금액의 분포를 확인해보기.

```
print(bc3['이용금액'].min(),bc3['이용금액'].max())
```

```
440 1418021926
```

440원과 14억은 너무 큰 차이
-> 고액결제와 일반결제로 나누어서 따로 해보기!

먼저 고액결제를 하는 사람들의 특징

```
bc3['고액금액분류'] = pd.cut(x=bc3['이용금액'],
                        bins=np.array([440,100000000,1000000000,1418021926]),
                        labels=['1억 이하','1억~10억','10억 이상'],
                        include_lowest=True)
print(bc3[['이용금액','고액금액분류']].head(20))
```

	이용금액	고액금액분류
0	7927440	1억 이하
1	274100	1억 이하
2	34395725	1억 이하
3	31860800	1억 이하
4	2546487	1억 이하
5	508970	1억 이하
6	3372800	1억 이하
7	93692161	1억 이하
8	14675900	1억 이하
9	6648030	1억 이하
10	490425014	1억~10억
11	13423621	1억 이하
12	5625955	1억 이하
13	11098894	1억 이하
14	3007800	1억 이하
15	61800	1억 이하
16	56176470	1억 이하
17	11396340	1억 이하
18	1535160	1억 이하
19	4959300	1억 이하

CONTENTS3 머신러닝(DecisionTree)

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
import warnings
warnings.filterwarnings('ignore')

X=bc3.drop(['이용금액','고액금액분류'],axis=1)
y=bc3['고액금액분류']

dt_clf_고액 = DecisionTreeClassifier(random_state=22)

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=22)

dt_clf_고액.fit(X,y)
```

```
DecisionTreeClassifier(random_state=22)
```

```
from sklearn.tree import export_graphviz

export_graphviz(dt_clf_고액,out_file='tree.dot',class_names=np.array(['1억 이하','1억~10억','10억 이상']),
                feature_names = list(X.columns),impurity=True,filled=True)
```

```
import graphviz
with open('tree.dot',encoding='UTF-8') as f:
    dot_graph=f.read()
graphviz.Source(dot_graph)
```

CONTENTS3 머신러닝(DecisionTree)

CONTENTS3 머신러닝(DecisionTree)

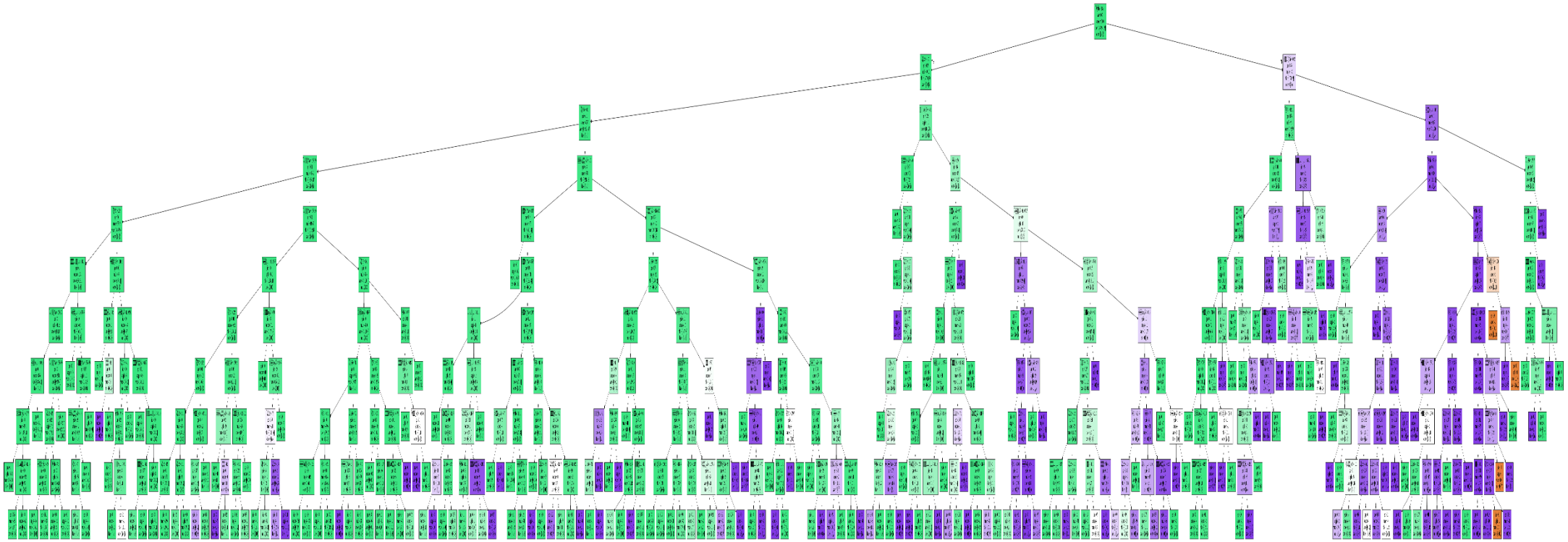
적절한 하이퍼 파라미터를 차아서 제한해주면 조금 더 간단한 결과를 얻을 수 있다.

```
import warnings
warnings.filterwarnings('ignore')
from sklearn.model_selection import GridSearchCV

parameters = {'max_depth': [5,6,7,8,9,10,11,12,13,14,15]}
dtree=DecisionTreeClassifier(random_state=22)
grid_dtree = GridSearchCV(dtree, param_grid=parameters, cv=3, refit=True)
grid_dtree.fit(X,y)
print('최적 하이퍼 파라미터: {0}'.format(grid_dtree.best_params_))
```

```
최적 하이퍼 파라미터: {'max_depth': 10}
```

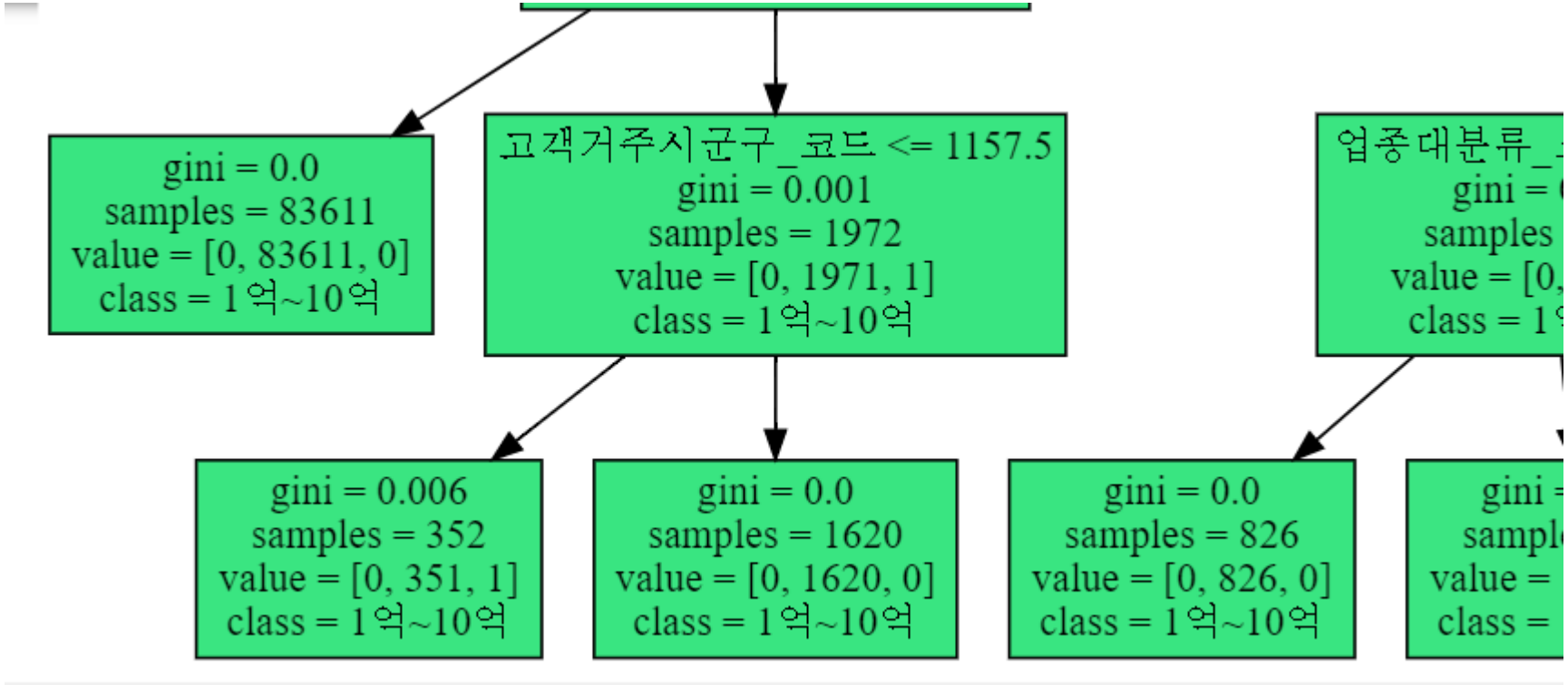
```
dt_clf_고액 = DecisionTreeClassifier(max_depth=10,random_state=22).fit(X,y)
export_graphviz(dt_clf_고액,out_file='tree_best.dot',class_names=np.array(['1억 이하','1억~10억','10억 이상']),
                feature_names = list(X.columns),impurity=True,filled=True)
(graph,)=pydot.graph_from_dot_file('tree_best.dot',encoding='UTF-8')
graph.write_png('tree_best.png')
with open('tree_best.dot',encoding='UTF-8') as f:
    dot_graph=f.read()
graphviz.Source(dot_graph)
```

CONTENTS3 머신러닝(DecisionTree)



CONTENTS3 머신러닝(DecisionTree)

각 조건에 따라 분류가 되면
소비 금액이 없는 상황에서도 특정 특징을 가진 사람들이
어느 정도의 소비를 한다는 예측을 할 수 있다.



CONTENTS3 머신러닝(DecisionTree)

고액 결제가 아닌 경우도 머신러닝해보기

```
bc3_=bc3.loc[bc3.이용금액<=100000000,:]
bc3_.drop(['고액금액분류'],axis=1,inplace=True)
bc3_['금액분류'] = pd.cut(x=bc3_['이용금액'],
                      bins=np.array([440,1000000,10000000,100000000]),
                      labels=['100만원 이하','1천만원 이하','1억 이하'],
                      include_lowest=True)
print(bc3_[['이용금액','금액분류']].head(20))
```

```
        이용금액      금액분류
0    7927440   1천만원 이하
1     274100  100만원 이하
2   34395725     1억 이하
3   31860800     1억 이하
4    2546487   1천만원 이하
5     508970  100만원 이하
6    3372800   1천만원 이하
7   93692161     1억 이하
8   14675900     1억 이하
9    6648030   1천만원 이하
11  13423621     1억 이하
12   5625955   1천만원 이하
13  11098894     1억 이하
14   3007800   1천만원 이하
15     61800  100만원 이하
16  56176470     1억 이하
```

CONTENTS3 머신러닝(DecisionTree)

처음부터 적절한 하이퍼 파라미터를 찾아서 머신러닝하기!

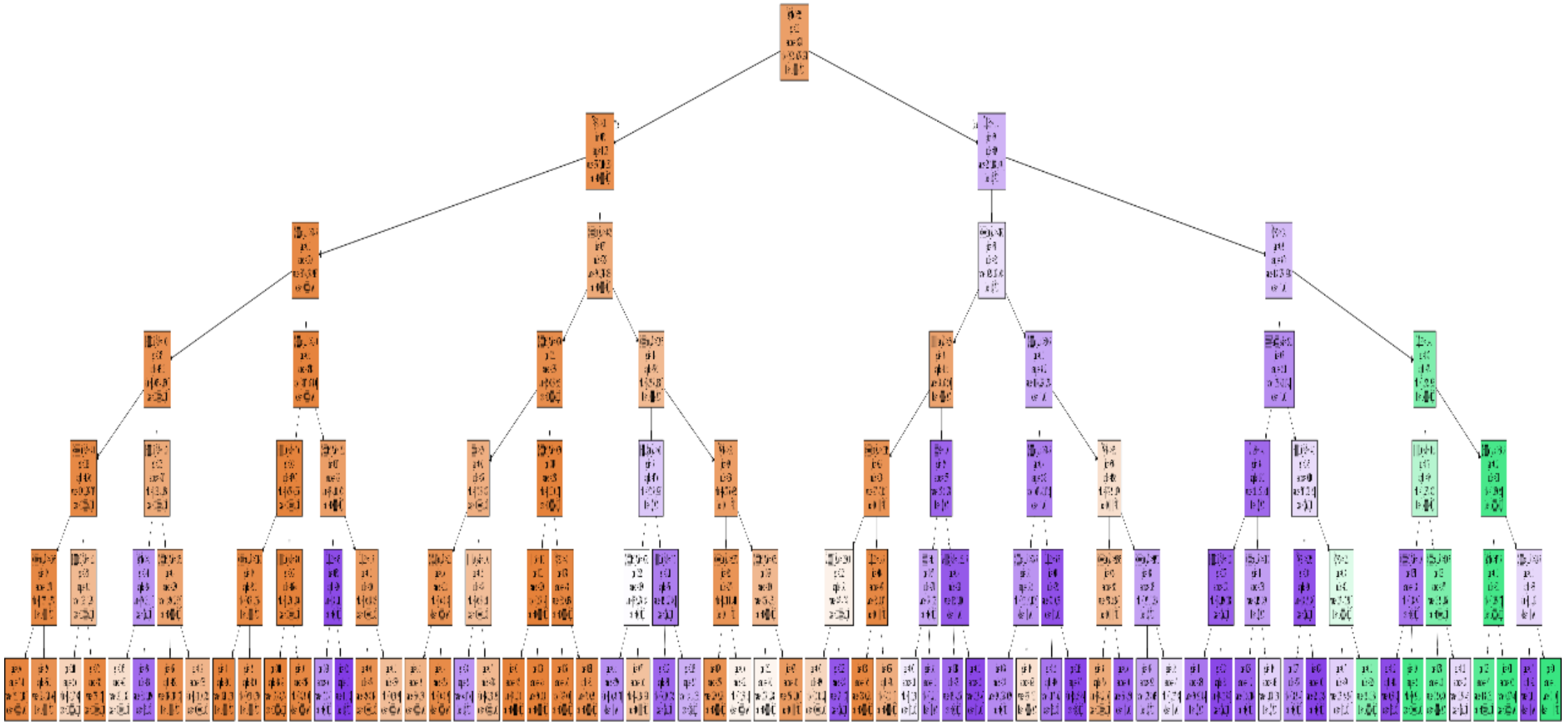
```
X=bc3_.drop(['이용금액','금액분류'],axis=1)
y=bc3_['금액분류']

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=22)
```

```
parameters = {'max_depth': [1, 2, 3, 4, 5, 6], 'min_samples_split':[1, 2, 3, 4, 5, 6]}
dtree=DecisionTreeClassifier(random_state=22)
grid_dtree = GridSearchCV(dtree, param_grid=parameters, cv=3, refit=True)
grid_dtree.fit(X,y)
print('최적 하이퍼 파라미터: {0}'.format(grid_dtree.best_params_))
```

```
최적 하이퍼 파라미터: {'max_depth': 6, 'min_samples_split': 2}
```

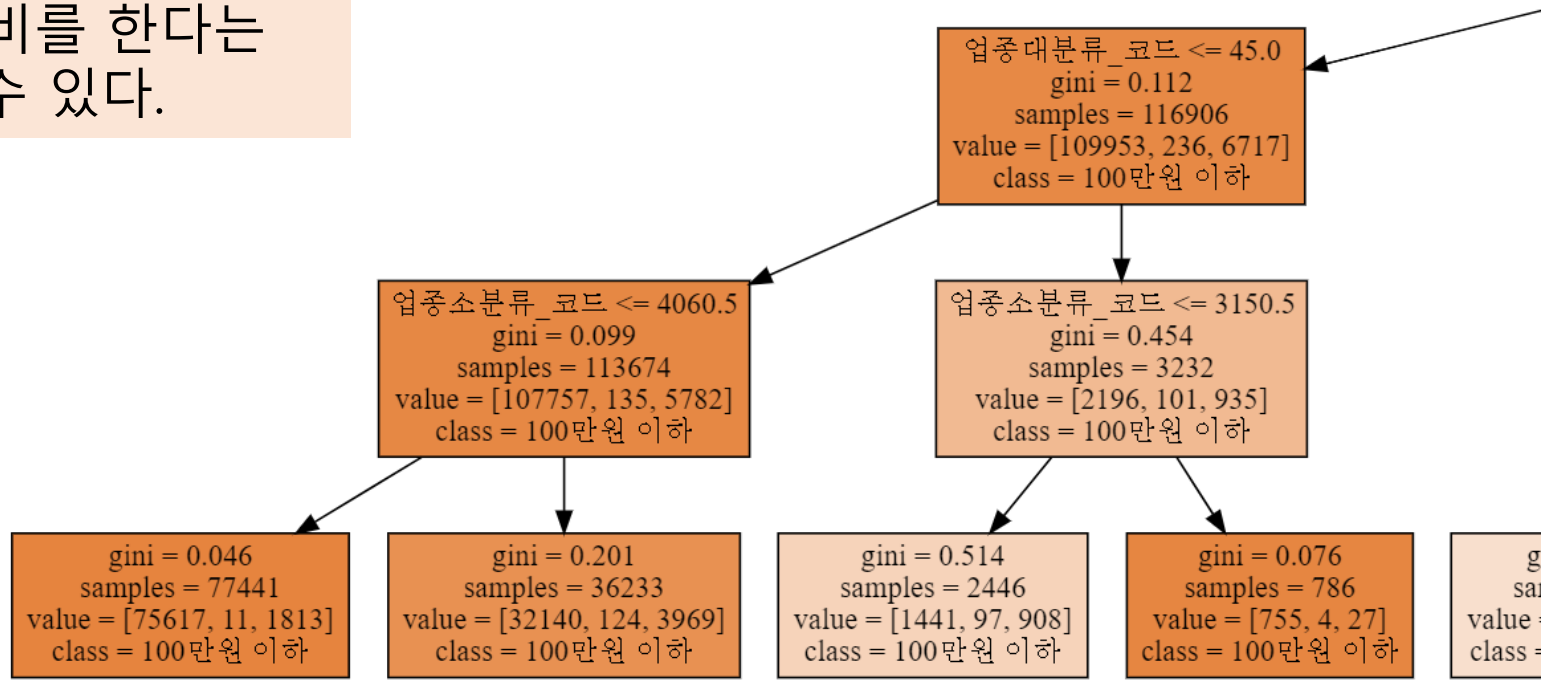
```
dt_clf = DecisionTreeClassifier(max_depth=6, min_samples_split=2,random_state=22).fit(X,y)
export_graphviz(dt_clf,out_file='tree1.dot',class_names=np.array(['100만원 이하','1천만원 이하','1억 이하']),
                feature_names = list(X.columns),impurity=True,filled=True)
(graph,)=pydot.graph_from_dot_file('tree1.dot',encoding='UTF-8')
graph.write_png('tree1.png')
with open('tree1.dot',encoding='UTF-8') as f:
    dot_graph=f.read()
graphviz.Source(dot_graph)
```

CONTENTS3 머신러닝(DecisionTree)



CONTENTS3 머신러닝(DecisionTree)

각 조건에 따라 분류가 되면
소비 금액이 없는 상황에서도
특정 특징을 가진 사람들이
어느 정도의 소비를 한다는
예측을 할 수 있다.



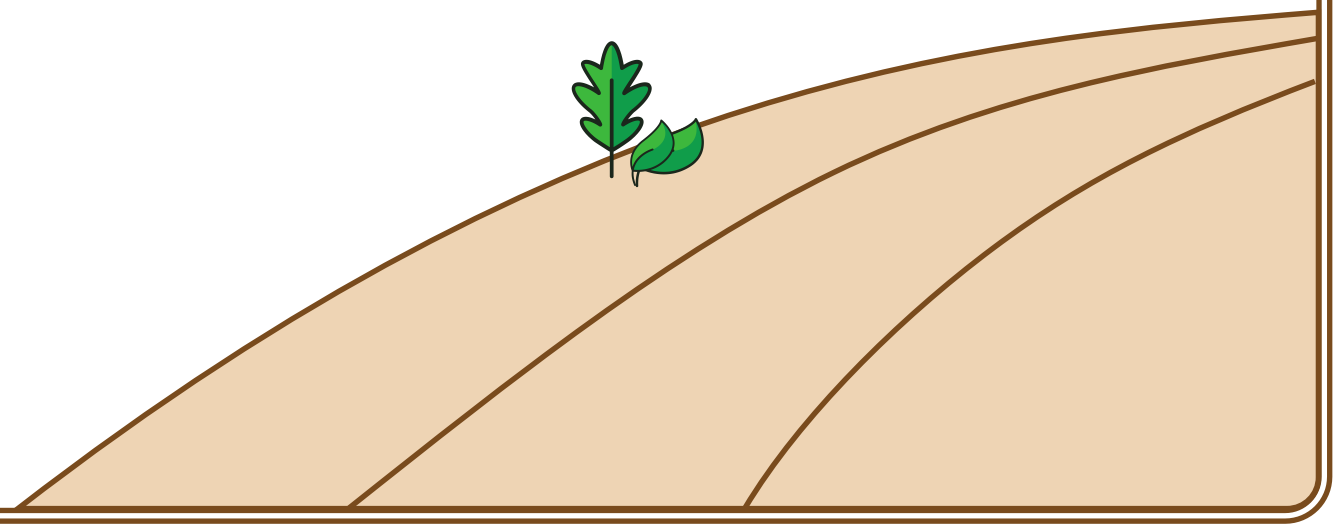
아쉬운점

일반적인 상식에서 크게 벗어나지 않는 분석 결과만 도출했다.

DecisionTree의 모양이 너무 복잡해서 한눈에 보기 어려웠다.
해당 문제를 파악하는 데 있어서 더 한눈에 알아보기 쉬운 방법을 고민해야한다.

Bc카드 내역에 대해 한 달간의 건수라는 너무 단기간의 데이터만 가지고 데이터 분석을 시도했다.

감사합니다.



18102002 이현진