프로젝트 기반 빅데이터 서비스 솔루션 개발 전문 과정

교과목명 : 머신러닝응용

- 평가일 : 03.13
- 성명 : 이재우
- 점수 : 75

문제 : LMEMBERS의 상품구매데이터를 이용하여 개인맞춤 상품 추천솔루션을 구축 후 다양한 활용 방안을 시현하세요.

In [1]:

```
import cx_Oracle
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from matplotlib import font_manager, rc
font_path = "C:/Windows/Fonts/NGULIM.TTF"
font = font_manager.FontProperties(fname=font_path).get_name()
rc('font', family=font)
con = cx_Oracle.connect('LM_PDB/LM@localhost:1521/xepdb1')
cursor = con.cursor()
```

In [2]:

```
cursor.execute("select 소분류코드, 중분류명, 소분류명 from prodcl")
x = cursor.fetchall()
df1 = pd.DataFrame(x,columns = ['소분류코드','중분류명','소분류명'])
df1
```

Out[2]:

	소분류코드	중분류명	소분류명
0	A060143	스포츠	레드페이스
1	A060144	스포츠	에코로바
2	A060145	스포츠	웨스트우드
3	A060146	스포츠	투스카로라
4	A060147	스포츠	아크테릭스
...	...	...	...
4381	A060249	골프용품	골프존마켓
4382	A060252	골프용품	해리토리 상품군
4383	A060107	스포츠	스포츠용품
4384	B710204	등산	등산소품
4385	B360102	롤러보드	스케이트보드

4386 rows × 3 columns

In [3]:

```
cursor.execute("select 고객번호, 소분류코드, count(*) from purprod group by 고객번호, 소분류코드
x = cursor.fetchall()
df2 = pd.DataFrame(x,columns = ['고객번호','소분류코드','구매횟수'])
df2
```

Out[3]:

	고객번호	소분류코드	구매횟수
0	00001	A010101	5
1	00001	A010103	3
2	00001	A010104	1
3	00001	A010106	3
4	00001	A010201	2
...	...	...	...
6527915	19383	D080203	5
6527916	19383	D080204	4
6527917	19383	D080205	1
6527918	19383	D080302	2

In [4]:

```
df2_matrix = df2.pivot_table('구매횟수',index = '고객번호', columns = '소분류코드')
df2_matrix
```

Out[4]:

소분류코드 / 고객번호	A010101	A010102	A010103	A010104	A010105	A010106	A010201	A010202	A010203	A010204	...
00001	5.0	NaN	3.0	1.0	NaN	3.0	2.0	5.0	NaN	NaN	...
00002	29.0	NaN	12.0	18.0	NaN	2.0	10.0	3.0	NaN	NaN	...
00003	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...
00004	15.0	NaN	8.0	2.0	NaN	NaN	7.0	NaN	NaN	NaN	...
00005	18.0	NaN	10.0	5.0	NaN	4.0	1.0	6.0	NaN	NaN	...
...	...	...	...	...	...	...	...	...	...	...	...
19379	NaN	NaN	4.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...

In [20]:

```
prod_df.소분류명.value_counts()
```

Out[20]:

```
어묵              23453
종량제봉투           22716
고추              22322
일반스낵            21877
양파              21128
                ...
장류세트               1
온라인기능헤어케어          1
규격고등어선물세트          1
기타세트               1
기타한방약재             1
Name: 소분류명, Length: 3520, dtype: int64
```

In [23]:

```
prod_df = pd.merge(df2,df1,on='소분류코드')

prod_matrix = prod_df.pivot_table('구매횟수',index = '고객번호', columns = '소분류명',aggfunc=

prod_matrix = prod_matrix.fillna(0)
prod_matrix
```

Out[23]:

소분류명	14K	2단우산	3단우산	3분요리류	4대B/D	5ON THE GO	ACC Bloom (1F)	ACC Bloom (3F)	AK골프	ANDZ	...	휴대폰기타용품	휴모니아	휴지류	휴지통	흑미	흑미류	흰다리새우	히터	히
고객번호																				
00001	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.
00002	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.
00003	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.
00004	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.
00005	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.

In [24]:

```
1 prod_matrix_T = prod_matrix.transpose()
2 prod_matrix_T
```

Out[24]:

고객번호 소분류명	00001	00002	00003	00004	00005	00006	00007	00008	00009	00010	...	19374	1937
14K	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.
2단우산	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.
3단우산	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.
3분요리류	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	2.0	0.
4대B/D	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	0.0	...	0.0	0.
...	...	...	...	...	...	...	...	...	...	...	...	...	.
흑미류	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	1.
흰다리새우	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	...	0.0	1.
히터	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.
히터기	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.
힐앤토트	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.

3520 rows × 19383 columns

In [25]:

```
from sklearn.metrics.pairwise import cosine_similarity

item_sim = cosine_similarity(prod_matrix_T,prod_matrix_T)

item_sim_df = pd.DataFrame(data = item_sim, index = prod_matrix.columns, columns = prod_matrix.

print(item_sim_df.shape)
item_sim_df
```

```
(3520, 3520)
```

In [26]:

```
def predict_prod_topsim(prod_arr,item_sim_arr,n=20):
    pred = np.zeros(prod_arr.shape)

    for col in range(prod_arr.shape[1]):

        top_n_times = [np.argsort(item_sim_arr[:,col])[:-n-1:-1]]

        for row in range(prod_arr.shape[0]):
            pred[row,col] = item_sim_arr[col,:][top_n_times].dot(prod_arr[row,:][top_n_times].T
            pred[row,col] /= np.sum(np.abs(item_sim_arr[col,:][top_n_times]))

    return pred

prod_pred = predict_prod_topsim(prod_matrix.values,item_sim_df.values,n=20)

prod_pred_matrix = pd.DataFrame(data = prod_pred, index = prod_matrix.index, columns = prod_mat

prod_pred_matrix
```

```
C:₩Users₩Master₩AppData₩Local₩Temp₩ipykernel_21256₩1731475551.py:9: FutureWarning:
Using a non-tuple sequence for multidimensional indexing is deprecated; use `arr[t
uple(seq)]` instead of `arr[seq]`. In the future this will be interpreted as an ar
ray index, `arr[np.array(seq)]`, which will result either in an error or a differe
nt result.
  pred[row,col] = item_sim_arr[col,:][top_n_times].dot(prod_arr[row,:][top_n_time
s].T)
C:₩Users₩Master₩AppData₩Local₩Temp₩ipykernel_21256₩1731475551.py:10: FutureWarnin
g: Using a non-tuple sequence for multidimensional indexing is deprecated; use `ar
r[tuple(seq)]` instead of `arr[seq]`. In the future this will be interpreted as an
array index, `arr[np.array(seq)]`, which will result either in an error or a diffe
rent result.
  pred[row,col] /= np.sum(np.abs(item_sim_arr[col,:][top_n_times]))
```

In [28]:

```
def get_zero_prod(prod_matrix,userId):
    user_prod = prod_matrix.loc[userId,:]

    already_buy = user_prod[user_prod > 0].index.tolist()

    prod_list = prod_matrix.columns.tolist()

    unbuy_list = [prod for prod in prod_list if prod not in already_buy]

    return unbuy_list

def recomm_prod_by_userid(pred_df, userId, unbuy_list, top_n=10):

    recomm_prod = pred_df.loc[userId,unbuy_list].sort_values(ascending=False)[:top_n]

    return recomm_prod
```

In [64]:

```
user_prod_id = prod_matrix.loc['00005',:]
user_prod_id[user_prod_id>0].sort_values(ascending=False)[:10]
```

Out[64]:

```
소분류명
유제품        160.0
채소          88.0
농산가공         75.0
밥류          47.0
일반가공식품       41.0
청과          33.0
수입식품         25.0
주류          24.0
일식델리         19.0
서양델리         18.0
Name: 00005, dtype: float64
```

In [65]:

```
unbuy_list = get_zero_prod(prod_matrix,'00005')

recomm_prod =recomm_prod_by_userid(prod_pred_matrix,'00005',unbuy_list,top_n=10)

recomm_prod_df = pd.DataFrame(data=recomm_prod.values,index=recomm_prod.index,columns=['pred_sc

recomm_prod_df
```

Out[65]:

	pred_score
소분류명	
계육	26.095324
가공행사	25.823045
건과	25.094209
멸치류	25.036357
전문베이커리	24.723370
수입육	23.952036
건생선	23.916558
떡	23.831161
규격김치	23.702259
선식(가루류)	23.282230

In [ ]:

```
# 활용방안
# 1.
```