CBR/AR/CF Report

짱짱조

[1] 범주형 타겟 변수에 대한 CBR 모델

1.1 모델 목적

해당 부분에는 과일의 무게와 색상값 데이터를 활용해서 해당 과일이 오렌지인지 자몽인지 분류하는 모델을 CBR을 활용해 생성해보기로 하였다.

1.2 데이터 전처리

범주형 타겟 변수에 대한 CBR 모델 실습에서는 “Orange and grapefruit diameter, weight, and color data”를 활용하였다. 총 6개의 columns와 10,000개의 rows로 이루어져 있으며, 각 row에는 Label을 부여하는 대신, “name” column에서 orange를 0으로, grapefruit를 1로 취급하여 처리하였다.

```
In [4]: df_cat_knn.dropna(axis=0,inplace = True)
        df_cat_knn['name'].replace({'orange':0,'grapefruit':1},inplace=True)
```

기존 데이터에서 “name” column만 분리하여 타겟 데이터(y_cat_knn)로 설정하고 나머지 데이터(x_cat_knn)를 표준화하였다.

```
In [9]: x_cat_knn = pd.DataFrame(StandardScaler().fit_transform(x_cat_knn), columns=x_cat_knn.columns, index = x_cat_knn.index)
        x_cat_knn
```

Out [9]:

	diameter	weight	red	green	blue
0	-3.601950	-3.022554	1.739978	0.767810	-1.033372
1	-3.114207	-2.978392	1.164848	0.169920	-0.923007
2	-2.852366	-2.734645	0.206299	0.426159	-1.033372
3	-2.826695	-2.719925	0.877283	0.426159	-0.812642
4	-2.821561	-2.714447	0.685574	-0.342557	-0.260815
...	...	...	...	...	...
9995	2.759248	2.698988	-0.464686	0.084507	0.953203
9996	2.790053	2.725690	-0.560540	-0.684208	-0.481546
9997	2.882467	2.788339	1.356558	0.511571	0.953203
9998	3.051894	2.912951	-1.135670	-0.342557	-0.040085
9999	3.324003	2.959851	-0.177121	-0.171731	-1.033372

10000 rows × 5 columns

이때 각 변수별로의 기초통계량은 다음과 같이 나타났다.

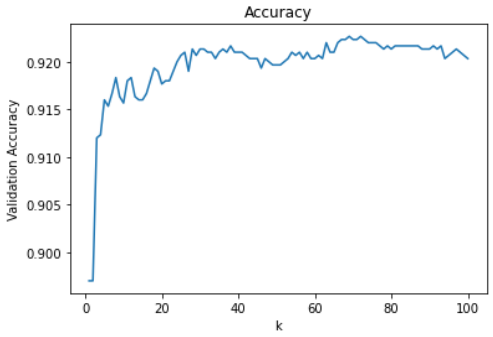
	diameter	weight	red	green	blue
count	1.000000e+04	1.000000e+04	1.000000e+04	1.000000e+04	1.000000e+04
mean	1.507550e-15	-3.791634e-16	-2.082490e-15	2.120967e-15	3.363820e-15
std	1.000050e+00	1.000050e+00	1.000050e+00	1.000050e+00	1.000050e+00
min	-3.601950e+00	-3.022554e+00	-3.723753e+00	-3.844481e+00	-1.033372e+00
25%	-7.781737e-01	-7.815911e-01	-6.563954e-01	-6.842078e-01	-1.033372e+00
50%	2.215381e-03	-2.252329e-03	1.458912e-02	-9.053757e-04	-1.504499e-01
75%	7.723362e-01	7.761450e-01	6.855736e-01	6.823970e-01	6.221069e-01
max	3.324003e+00	2.959851e+00	3.657076e+00	3.415607e+00	4.926352e+00

1.3 CBR 모델 생성, 학습 및 평가

본격적으로 데이터셋을 분리하기 전에 행 순서 섞는 shuffling을 해주었다. 이렇게 섞은 뒤에 x_cat_knn과 y_cat_knn을 train 데이터와 test 데이터 7:3의 비율로 분리한 후 sklearn.neighbors.KNeighborsClassifier 클래스를 이용하여 근접수(k)를 3으로 설정하여 train 데이터를 학습시켰다. 학습이 끝나면 모델의 성능을 평가하기 위해 category를 예측시키고 accuracy를 계산해보면 0.912가 나온다.

1.4 근접수(k)를 달리한 CBR 모델 비교

위와 같은 방법으로 k를1~100의 범위에서 달리하여 CBR 모델을 생성하고 train 데이터로 학습시킨 후 test 데이터로 accuracy를 측정해보면 다음과 같은 결과를 얻을 수 있다.




```
In [23]: accuracies.index(max(accuracies))
Out[23]: 68
```

위 그래프에서 k를 69로 설정한 모델의 accuracy가 가장 높다는 사실을 확인할 수 있다. 이를 바탕으로 CBR 모델을 생성하여 accuracy를 구해보면 0.9226666666666666이라는 값이 나와 k를 3으로 설정할 때보다 높은 정확도를 보였다.

1.5 Confusion Matrix 적용

범주형 데이터의 모델 성능을 평가하기 위한 지표로 Confusion Matrix도 계산해보았다.

```
In [27]: # Confusion Matrix 방법1
from sklearn.metrics import confusion_matrix
confusion_matrix(validation_labels, classifier.predict(validation_data))

Out[27]: array([[1388,  126],
       [ 106, 1380]], dtype=int64)

In [28]: # Confusion Matrix 방법2
import numpy as np
import pandas as pd
y_actu = pd.Series(np.array(validation_labels), name='Actual') # validation_labels가 dataframe형태이므로 array형태로
y_pred = pd.Series(classifier.predict(validation_data), name='Predicted')
df_confusion = pd.crosstab(y_actu, y_pred)

df_confusion

Out[28]:
```

Predicted	0	1
Actual		
0	1388	126
1	106	1380

Matrix를 보면 정확히 예측한 데이터는 1388+1380=2768개이고, 제대로 못한 데이터는 106+126=232개이다. 따라서 정확도는 2768/3000=0..9226666666666666으로 이전에 보였던 accuracy와 일치하였다.

[2] 연속형 타겟 변수에 대한 CBR 모델

2.1 모델 목적

해당 부분에는 야구선수 타자들의 개인 성적을 토대로 그 선수의 연봉을 예측해보는 모델을 CBR 기법을 통해서 생성해보기로 했다.

2.2 데이터 전처리

연속형 타겟 변수에 대한 CBR 모델 실습에서는 한국 프로야구 타자 데이터를 이용하였다. 총 37개의 column, 1913개의 row가 있다. 데이터를 pandas.Dataframe 형식으로 바꿔 첫 5 row만 출력하면 다음과 같다.

	batter_name	age	G	PA	AB	R	H	2B	3B	HR	...	tp	1B	FBP	avg	OBP	SLG	OPS	p_year	YAB	YOPS
0	백용환	24.0	26.0	58.0	52.0	4.0	9.0	4.0	0.0	0.0	...	포수	5.0	6.0	0.173	0.259	0.250	0.509	2014	79.0	0.580
1	백용환	25.0	47.0	86.0	79.0	8.0	14.0	2.0	0.0	4.0	...	포수	8.0	5.0	0.177	0.226	0.354	0.580	2015	154.0	0.784
2	백용환	26.0	65.0	177.0	154.0	22.0	36.0	6.0	0.0	10.0	...	포수	20.0	20.0	0.234	0.316	0.468	0.784	2016	174.0	0.581
3	백용환	27.0	80.0	199.0	174.0	12.0	34.0	7.0	0.0	4.0	...	포수	23.0	20.0	0.195	0.276	0.305	0.581	2017	17.0	0.476
4	백용환	28.0	15.0	20.0	17.0	2.0	3.0	0.0	0.0	0.0	...	포수	3.0	3.0	0.176	0.300	0.176	0.476	2018	47.0	0.691

5 rows × 37 columns

같은 선수에 대하여, 연도별로 row가 존재한다는 것을 알 수 있다. 이 데이터에서 타자 이름이나 생년과 같이 분석에 불필요하거나, 타석 위치나 최근 포지션과 같이 범주형으로 표현되거나, 또는 결측치를 포함하는 column 5개를 제외하여 최종적으로 32개의 column을 사용하였다.

전처리를 마친 데이터에서 타자의 연봉을 나타내는 'salary' column을 y로 지정하면 다음과 같다.

```
0        2500
1        2900
2        6000
3        6000
4        5500
        ...
1908    30000
1909     3100
1910     6200
1911    50000
1912    50000
Name: salary, Length: 1885, dtype: int64
```

전처리를 마친 데이터에서 y로 지정한 'salary' column을 제외한 나머지를 sklearn.preprocessing. StandardScaler 클래스로 표준화하여 x로 지정하면 다음과 같다.

	age	G	PA	AB	R	H	2B	3B	HR	TB	...	war	1B	FBP
0	-0.646836	-1.243421	-1.064360	-1.059032	-1.047453	-1.059278	-0.736921	-0.630259	-0.760842	-1.005641	...	-0.738968	-1.109647	-0.902486
1	-0.426071	-0.763082	-0.920722	-0.899918	-0.909964	-0.964578	-0.933663	-0.630259	-0.289109	-0.828181	...	-0.947885	-1.028197	-0.943617
2	-0.205306	-0.351363	-0.453899	-0.457938	-0.428751	-0.547901	-0.540179	-0.630259	0.418491	-0.307634	...	-0.285413	-0.702398	-0.326649
3	0.015459	-0.008264	-0.341041	-0.340076	-0.772475	-0.585780	-0.441808	-0.630259	-0.289109	-0.532416	...	-0.928400	-0.620948	-0.326649
4	0.236225	-1.495027	-1.259297	-1.265289	-1.116198	-1.172917	-1.130404	-0.630259	-0.760842	-1.123947	...	-0.779561	-1.163947	-1.025879
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1908	1.119286	1.089654	1.423653	1.457313	1.358609	1.497609	1.820723	-0.064796	1.597825	1.632588	...	1.084995	1.279546	1.277466
1909	0.236225	-0.282743	-0.207663	-0.239894	-0.016284	-0.244862	-0.441808	-0.064796	-0.760842	-0.437771	...	-0.592835	-0.050800	0.002400
1910	0.456990	-1.472154	-1.290077	-1.288862	-1.047453	-1.210797	-1.130404	-0.630259	-0.760842	-1.147608	...	-0.855875	-1.218247	-1.149273
1911	0.015459	-0.236996	0.171952	0.278697	0.086833	0.418034	0.541901	0.500668	0.536425	0.508679	...	0.570822	0.302149	-0.491174
1912	0.236225	1.249767	1.608331	1.716609	2.286662	1.819587	2.410948	-0.630259	2.305425	2.093983	...	2.189657	1.442446	0.866155

1885 rows × 31 columns

다음은 해당 데이터셋의 기초통계량 관련 지표이다. (변수가 너무 많은 관계로 기타 부분은 코드 참조)

	age	G	PA	AB	R	H	2B	3B	HR	TE
count	1.885000e+03	1.885000e+03	1.885000e+03	1.885000e+03	1.885000e+03	1.885000e+03	1.885000e+03	1.885000e+03	1.885000e+03	1.885000e+03
mean	3.656815e-16	1.173244e-16	-8.716870e-18	6.090030e-17	1.104922e-16	1.258056e-16	5.088768e-17	-6.404544e-16	-2.563231e-16	-6.431637e-17
std	1.000265e+00	1.000265e+00	1.000265e+00	1.000265e+00	1.000265e+00	1.000265e+00	1.000265e+00	1.000265e+00	1.000265e+00	1.000265e+00
min	-1.971428e+00	-1.815253e+00	-1.356766e+00	-1.359579e+00	-1.184943e+00	-1.229737e+00	-1.130404e+00	-6.302594e-01	-7.608422e-01	-1.159439e+00
25%	-6.468362e-01	-8.774484e-01	-1.013061e+00	-1.000101e+00	-9.099640e-01	-9.645785e-01	-9.336625e-01	-6.302594e-01	-7.608422e-01	-9.346569e-01
50%	1.545942e-02	2.662159e-01	-3.324518e-02	-3.363588e-02	-1.881452e-01	-1.122830e-01	-1.466953e-01	-6.302594e-01	-4.070421e-01	-1.893276e-01
75%	6.777551e-01	8.837946e-01	9.106610e-01	9.092568e-01	7.055353e-01	8.347121e-01	7.386428e-01	5.006677e-01	3.005580e-01	7.689528e-01
max	3.106172e+00	1.455627e+00	2.085414e+00	2.028942e+00	3.455321e+00	2.577183e+00	3.493028e+00	8.982621e+00	5.607559e+00	3.253384e+00

8 rows × 31 columns

2.3 CBR 모델 생성, 학습 및 평가

이렇게 지정한 x와 y를 train 데이터와 test 데이터 7:3의 비율로 분리한 후, sklearn.neighbors. KNeighborsRegressor 클래스를 이용하여 train 데이터를 학습시킨다. 이 때, 근접수(K)는 3으로 설정하였다. 학습이 끝나면 모델의 성능을 평가하기 위해 test 데이터에 대한 y 값을 예측시키고 이에 대해 mean square error를 산출하면 265266303.95771104이라는 값이 도출된다.

2.4 근접수(K)를 달리한 CBR 모델 비교

위와 같은 방법으로 K를 1~20의 범위에서 달리하여 CBR 모델을 생성하고 train 데이터로 학습시킨 후 test 데이터로 MSE를 측정해보면 다음과 같은 그래프를 그릴 수 있다.

위 그래프에서 K를 9로 설정한 모델의 MSE가 가장 낮다는 사실을 확인할 수 있다. 이처럼 K가 무조건 크거나 작다고 해당 모델이 최적의 결과를 도출하지는 않는다. 이를 바탕으로 K를 9로 설정한 CBR 모델을 생성하여 MSE를 구해보면 235114252.02818453이라는 값이 나오고 이는 K를 3으로 설정했던 최초 모델보다 낮은 값임을 확인할 수 있다.

[3] AR 모델

3.1 모델 목적

해당 부분에서는 한 India 식당에서의 주문 데이터를 통해 상품들의 Association을 알아보는 모델을 생성해보았다.

3.2 데이터 전처리

	Order Number	Order Date	Item Name	Quantity	Product Price	Total products
0	16118	03/08/2019 20:25	Plain Papadum	2	0.80	6
1	16118	03/08/2019 20:25	King Prawn Balti	1	12.95	6
2	16118	03/08/2019 20:25	Garlic Naan	1	2.95	6
3	16118	03/08/2019 20:25	Mushroom Rice	1	3.95	6
4	16118	03/08/2019 20:25	Paneer Tikka Masala	1	8.95	6

Association Rule 실습에서는 kaggle의 'Takeaway Food Orders' 데이터를 이용하였다. 데이터는 아래와 같다. 총 6개의 column, 74,818의 row로 이루어져 있으며 데이터는 위와 같다. 이 중 Order Number와 Item Name column을 활용해 Item 주문 간의 연관규칙을 분석하고자 한다.

	Aloo Chaat	Aloo Gobi	Aloo Methi	Baingan Hari Mirch	Bengal Fish Biryani	Bengal Fish Karahi	Bengal Fry Fish	Bengal King Prawn	Bengal Salad	Bhindi Bhajee	Bhuna
0	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False
2	True	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...
13392	False	False	False	False	False	False	False	False	False	False	False
13393	False	False	False	False	False	False	False	False	False	False	False
13394	False	False	False	False	False	False	False	False	False	False	False
13395	False	False	False	False	False	False	False	False	False	False	False
13396	False	False	False	False	False	False	False	False	False	False	False

13397 rows × 248 columns

연관분석을 수행하기 위해 Order별로 주문한 Item들을 모아 order list를 array로 표현하고 mlxtend.preprocessing 모듈의 TransactionEncoder를 활용해 이를 One hot Encoding 해준다. 그 결과, 248개의 메뉴에 대해 총 13,397개의 주문이 Encoding 된 것을 확인할 수 있다.

3.3 Frequent Item-set Generation

	support	itemsets
0	0.130776	(Bombay Aloo)
1	0.073151	(Butter Chicken)
2	0.087333	(Chapati)
3	0.051653	(Chicken Tikka)
4	0.060088	(Chicken Tikka (Main))
5	0.159215	(Chicken Tikka Masala)
6	0.196163	(Garlic Naan)
7	0.101665	(Keema Naan)
8	0.087482	(Korma)
9	0.070389	(Korma - Chicken)
10	0.058819	(Madras)
11	0.154438	(Mango Chutney)

위의 그림은 Frequent Itemset을 만들어내는 방법 중 하나인 Apriori 알고리즘을 활용해 만들어 낸 frequent Itemset이다. Apriori Algorithm은 Transaction Data와 Minimum Support가 주어졌을 때, 먼저 transaction Data를 스캔하면서 1번 frequent itemsets L1을 구한다. 이후 k번째 frequent itemsets을 사용해 k+1번째 frequent itemsets를 구해주는데 이때 Join Step과 Prune Step을 통해 itemset 후보를 구하고 min support 조건을 만족하는 후보만 도출해낸다. 더 이상 k+1번째 frequent itemset이 발견되지 않을 때까지 위의 과정을 반복하며 Frequent Itemset을 찾아낸다.

실습에서는 mlxtend.frequent_patterns 모듈의 apriori 함수를 활용해 frequent Itemset을 구해주었다.

3.4 Association 분석 결과

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
13	(Mango Chutney)	(Mint Sauce)	0.154438	0.109204	0.053743	0.347994	3.186656	0.036878	1.366240
12	(Mint Sauce)	(Mango Chutney)	0.109204	0.154438	0.053743	0.492139	3.186656	0.036878	1.664950
43	(Pilau Rice, Mango Chutney)	(Plain Papadum)	0.073524	0.268418	0.058147	0.790863	2.946382	0.038412	3.498097
46	(Plain Papadum)	(Pilau Rice, Mango Chutney)	0.268418	0.073524	0.058147	0.216630	2.946382	0.038412	1.182679
42	(Pilau Rice, Plain Papadum)	(Mango Chutney)	0.133239	0.154438	0.058147	0.436415	2.825832	0.037570	1.500327
47	(Mango Chutney)	(Pilau Rice, Plain Papadum)	0.154438	0.133239	0.058147	0.376510	2.825832	0.037570	1.390177
29	(Onion Chutney)	(Plain Papadum)	0.077032	0.268418	0.058297	0.756783	2.819416	0.037620	3.007937
28	(Plain Papadum)	(Onion Chutney)	0.268418	0.077032	0.058297	0.217186	2.819416	0.037620	1.179038

Support값과 Confidence가 각각 최소 0.05와 0.1이라는 조건을 만족하는 항목들을 lift 내림차순으로 정렬한 결과이다. 위의 항목들을 모두 조건문장으로 나타내면 아래와 같다.

1,2) Mango Chutney와 Mint Sauce를 함께 주문한다.
- lift의 값은 3.187으로 신뢰도가 0.1을 만족하는 항목들 중 가장 높은 향상도 값을 가지며 Mango Chutney를 구매했을 때 Mint Sauce는 높은 확률로 같이 구매되고 그 반대도 성립하는 것을 알 수 있다.
- Mango Chutney가 antecedents일 때의 Confidence값은 0.348, Mint Sauce가 antecedents일 경우의 Confidence값은 0.492값으로 살짝 더 낮은 것을 확인할 수 있다.

3,4) Pilau Rice, Mango Chutney와 Plain Papadum를 함께 주문한다.
- lift의 값은 2.94로 Pilau Rice, Mango Chutney를 구매했을 때 Plain Papadum이 높은 확률로 같이 구매되고 그 반대도 성립하는 것을 의미한다.
- Pilau Rice(PR), Mango Chutney(MC)가 antecedents일 때의 confidence의 값은 0.791, Plain Papadum(PP)이 antecedents일 때의 confidence의 값은 0.217으로 Plain Papadum이 antecedents일 때가 더 낮은 신뢰도를 보이는 것을 확인할 수 있다. 이는 Pr(PRMC)의 값이 Pr(PP)의 값보다 작기 때문인 것으로 분석 가능하다.

5,6) Pilau Rice, Plain Papadum와 Mango Chutney를 함께 주문한다.
- lift의 값은 2.826으로 Pilau Rice, Plain Papadum을 구매했을 때 Plain Papadum이 높은 확률로 같이 구매되고 그 반대도 성립하는 것을 의미한다.
- Pilau Rice(PR), Plain Papadum(PP)이 antecedents일 때의 confidence의 값은 0.436, Mango Chutney가 antecedents일때는 0.377로 Mango Chutney일 때가 살짝 더 낮은 것을 확인할 수 있다.

7,8) Onion Chutney와 Plain Papadum을 함께 주문한다.
- lift값은 2.819로 Onion Chutney를 구매했을 때 Plain Papadum이 높은 확률로 같이 구매되고 그 반대도 성립하는 것을 의미한다.
- Onion Chutney(OC)가 antecedent일 때의 confidence 값은 0.757, Plain Papadum(PP)가 antecedent일 때의 confidence 값은 0.217로 PP가 antecedents일 때가 더 낮은 신뢰도를 보이는 것을 알 수 있다. 이는 Pr(OC)의 값이 Pr(PP)의 값보다 작기 때문인 것으로 분석 가능하다.

[4] CF 모델

4.1 모델 목적

해당 부분에서는 사용자들의 영화 평점 데이터를 통해 기존 선호 영화와 유사한 영화를 추천해주는 CF 모델을 생성해보았다.

4.2 데이터 전처리

	userId	movieId	rating	timestamp
0	1	31	2.5	1260759144
1	1	1029	3.0	1260759179
2	1	1061	3.0	1260759182
3	1	1129	2.0	1260759185
4	1	1172	4.0	1260759205
...	...	...	...	...
99999	671	6268	2.5	1065579370
100000	671	6269	4.0	1065149201
100001	671	6365	4.0	1070940363
100002	671	6385	2.5	1070979663
100003	671	6565	3.5	1074784724

100004 rows × 4 columns

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy
...	...	...	...
9120	162672	Mohenjo Daro (2016)	Adventure|Drama|Romance
9121	163056	Shin Godzilla (2016)	Action|Adventure|Fantasy|Sci-Fi
9122	163949	The Beatles: Eight Days a Week - The Touring Y...	Documentary
9123	164977	The Gay Desperado (1936)	Comedy
9124	164979	Women of '69, Unboxed	Documentary

9125 rows × 3 columns

→ ratings.csv 테이블(왼), movies.csv 테이블(오)

Collaborative Filtering 실습에서는 kaggle의 'Movielens (Small)' 데이터 중 사용자별 영화 평점을 나타낸 'ratings.csv' 데이터와 영화에 대한 정보를 나타낸 'movies.csv' 데이터를 활용하였다. 위의 테이블은 왼쪽부터 각각 'ratings.csv'와 'movies.csv'를 나타낸 것이다. 'ratings.csv'는 총 100,004 개의 row와 'userId', 'movieId', 'rating' column으로 이루어져있다. 아래의 테이블처럼 두 데이터를 'movieId'를 기준으로 merge한 data를 활용해 User-based CF와 Item-based CF를 수행해 사용자에게 영화를 추천하고 두 결과를 비교해보고자 한다.

	userId	movieId	rating	timestamp	title	genres
0	1	31	2.5	1260759144	Dangerous Minds (1995)	Drama
1	7	31	3.0	851868750	Dangerous Minds (1995)	Drama
2	31	31	4.0	1273541953	Dangerous Minds (1995)	Drama
3	32	31	4.0	834828440	Dangerous Minds (1995)	Drama
4	36	31	3.0	847057202	Dangerous Minds (1995)	Drama
...	...	...	...	...	...	...
99999	664	64997	2.5	1343761859	War of the Worlds (2005)	Action|Sci-Fi
100000	664	72380	3.5	1344435977	Box, The (2009)	Drama|Horror|Mystery|Sci-Fi|Thriller
100001	665	129	3.0	995232528	Pie in the Sky (1996)	Comedy|Romance
100002	665	4736	1.0	1010197684	Summer Catch (2001)	Comedy|Drama|Romance
100003	668	6425	1.0	993613478	6th Man, The (Sixth Man, The) (1997)	Comedy

100004 rows × 6 columns

→ ratings.csv와 movies.csv를 movieId를 기준으로 merge한 테이블

Collaborative Filtering을 진행하려면 사용자별 각 영화에 대한 평점을 나타내는 Preference Matrix가 필요하다. 따라서, 위의 merge한 dataframe에서 pivot_table 함수를 활용해 User의 평점을 Movie별로 나타내 User의 Preference를 나타냈다. 사용자별로 평점이 없는 영화의 경우 그 값을 0으로 표기했다. 그 결과 총 671명의 사용자에 대해 9,064개의 영화에 대한 평점을 담은 Rating(Preference) Matrix를 얻게 되었다.

title userId	"Great Performances" Cats (1998)	$9.99 (2008)	'Hellboy': The Seeds of Creation (2004)	'Neath the Arizona Skies (1934)	'Round Midnight (1986)	'Salem's Lot (2004)	'Til There Was You (1997)	'burbs, The (1989)	'night Mother (1986)	(500) Days of Summer (2009)	*batteries not included (1987)	...And God Spoke (1993)
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
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...
667	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
668	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
669	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
670	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
671	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

671 rows × 9064 columns

→ Rating(Preference) Matrix

4.3 User-based Collaborative Filtering

1) User Similiarity Matrix

User-based Collaborative Filtering은 Target user와 유사한 성향을 지닌 user들을 기반으로 item을 추천해주는 방법이다. 이때, sklearn.metrics.pairwise의 cosine_similarity를 활용해 User들 간의 Similarity를 구했다.

userId	1	2	3	4	5	6	7	8	9	10	11
1	0.000000	0.000000	0.000000	0.074482	0.016818	0.000000	0.083884	0.000000	0.012843	0.000000	0.000000
2	0.000000	0.000000	0.124295	0.118821	0.103646	0.000000	0.212985	0.113190	0.113333	0.043213	0.049610
3	0.000000	0.124295	0.000000	0.081640	0.151531	0.060691	0.154714	0.249781	0.134475	0.114672	0.092004
4	0.074482	0.118821	0.081640	0.000000	0.130649	0.079648	0.319745	0.191013	0.030417	0.137186	0.018987
5	0.016818	0.103646	0.151531	0.130649	0.000000	0.063796	0.095888	0.165712	0.086616	0.032370	0.020693
...	...	...	...	...	...	...	...	...	...	...	...
667	0.000000	0.425462	0.124562	0.088735	0.058252	0.000000	0.232051	0.069005	0.066412	0.032653	0.031539
668	0.000000	0.084646	0.124911	0.068483	0.042926	0.019563	0.058773	0.112366	0.194493	0.098561	0.055004
669	0.062917	0.024140	0.080984	0.104309	0.038358	0.024583	0.073151	0.055143	0.029291	0.060549	0.025804
670	0.000000	0.170595	0.136606	0.054512	0.062642	0.019465	0.096240	0.247687	0.384429	0.158650	0.043783
671	0.017466	0.113175	0.170193	0.211609	0.225086	0.087705	0.268672	0.406414	0.168497	0.189703	0.060489

671 rows × 671 columns

→ User to User similiarity matrix

이때, cosine similarity 계산시 주대각선의 값은 원래 1이지만 collatborative filtering은 recommendation이 목적이기 때문에 이미 구매한 항목에 대한 영향은 최대한으로 줄여 주대각선의 값을 모두 0으로 조정했다.

2) User Preference 예측

title	"Great Performances" Cats (1998)	$9.99 (2008)	'Hellboy': The Seeds of Creation (2004)	'Neath the Arizona Skies (1934)	'Round Midnight (1986)	'Salem's Lot (2004)	'Til There Was You (1997)	'burbs, The (1989)	'night Mother (1986)	(500) Days of Summer (2009)
userId										
1	0.107964	0.314610	0.031052	0.076373	0.307066	0.054340	0.630042	2.226901	0.825275	3.168993
2	0.109964	0.680772	0.350354	0.019059	0.408878	0.613119	0.965106	7.188990	1.889119	12.664465
3	0.062413	1.132310	0.343030	0.025862	0.362168	0.600303	0.702561	7.518544	1.383416	21.863861
4	0.205789	1.037894	0.640205	0.020235	0.662227	1.120358	2.034352	13.575863	3.777436	20.056962
5	0.260079	0.799233	0.372303	0.012113	0.575269	0.651531	1.164703	9.346513	1.901490	25.578486
...	...	...	...	...	...	...	...	...	...	...
667	0.079288	0.556467	0.301168	0.048465	0.472744	0.527043	0.920144	5.585534	1.508148	9.568899
668	0.084523	0.985344	0.206650	0.084523	0.433886	0.361638	0.589450	4.893544	1.407832	12.624461
669	0.180127	0.503276	0.174429	0.000000	0.255951	0.305250	0.968378	4.231660	1.194609	8.816636
670	0.179119	0.721603	0.338330	0.033640	0.452857	0.592077	0.775592	7.142516	1.276934	15.408412
671	0.125236	1.406633	0.665517	0.036342	0.684815	1.164654	1.447551	12.505059	2.254251	31.931692

→ 각 movie에 대한 user의 rating 예측

User들이 평점을 매기지 않은 영화들에 대해서도 평점을 예측해 User들이 선호할 영화를 추천하고자 한다. 위의 Matrix는 User들 간의 Similarity Matrix(671x671)와 Rating Matrix(671x9064)를 곱해 각 movie에 대한 user의 평점을 예측한 것이다. 이 예측된 평점이 높은 영화를 추천해주는 것이 사용자의 선호에 맞는 영화를 추천해주는 것이라는 추론이 가능하다.

4.4 Item-based Collaborative Filtering

1) Item Similiarity Matrix

Item-based Collaborative Filtering은 Target user가 선호하는 item을 기반으로 가장 유사한 성향의 item을 추천해주는 방법이다. 이를 수행하기 위해 sklearn.metrics.pairwise의 cosine_similarity를 활용해 Item들 간의 Similarity를 구했다. 이때, Item similarity matrix와 마찬가지로cosine similarity 계산시 주대각선의 값은 모두 0으로 조정했다.

title	"Great Performances" Cats (1998)	$9.99 (2008)	'Hellboy': The Seeds of Creation (2004)	'Neath the Arizona Skies (1934)	'Round Midnight (1986)	'Salem's Lot (2004)	'Til There Was You (1997)	'burbs, The (1989)	'night Mother (1986)	(500) Days of Summer (2009)	*batteries not included (1987)	...And God Spoke (1993)
title												
"Great Performances" Cats (1998)	0.000000	0.000000	0.0	0.164399	0.020391	0.0	0.014046	0.000000	0.000000	0.003166	0.000000	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...
xXx (2002)	0.000000	0.013899	0.0	0.000000	0.000000	0.0	0.000000	0.123940	0.000000	0.144961	0.112392	0.0
xXx: State of the Union (2005)	0.000000	0.000000	0.0	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.134815	0.000000	0.0
¡Three Amigos! (1986)	0.000000	0.058218	0.0	0.000000	0.000000	0.0	0.081620	0.331663	0.214498	0.064908	0.094151	0.0
À nous la liberté (Freedom for Us) (1931)	0.000000	0.000000	0.0	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
İtirazım Var (2014)	0.000000	0.000000	0.0	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.134815	0.000000	0.0

9064 rows × 9064 columns

→ Item to Item similiarity matrix

2) User Preference 예측

title	"Great Performances" Cats (1998)	$9.99 (2008)	'Hellboy': The Seeds of Creation (2004)	'Neath the Arizona Skies (1934)	'Round Midnight (1986)	'Salem's Lot (2004)	'Til There Was You (1997)	'burbs, The (1989)	'night Mother (1986)	(500) Days of Summer (2009)
userId										
1	0.327789	1.900466	0.427335	1.269944	4.806446	0.427335	4.936799	7.718179	6.104138	5.209114
2	0.760254	5.098898	8.874378	0.255822	15.196424	8.874378	19.381841	37.951869	27.640191	23.506391
3	0.180282	4.955293	5.778490	0.287800	9.279056	5.778490	10.272687	26.157646	14.651951	31.074348
4	3.634181	21.331482	37.415858	1.056453	57.049140	37.415858	88.691893	163.267154	127.263968	94.821523
5	2.574633	11.061546	12.961940	0.597115	28.393780	12.961940	18.782058	64.978770	25.878727	76.215718
...	...	...	...	...	...	...	...	...	...	...
667	0.433575	4.062333	7.496071	0.703973	15.797639	7.496071	16.669374	28.851621	20.590526	18.490218
668	0.115732	2.914094	2.512819	0.703973	5.949251	2.512819	4.512752	10.839869	6.767648	10.829945
669	0.809937	2.994675	2.812019	0.000000	8.804449	2.812019	12.919586	17.282733	14.416552	14.578043
670	0.707592	2.803251	5.235298	0.384195	8.906506	5.235298	7.638379	19.886886	9.232000	16.666334
671	1.161170	13.203950	22.415471	1.022262	33.087612	22.415471	25.504410	75.746802	30.808784	79.027782

671 rows × 9064 columns

→ 각 movie에 대한 user의 rating 예측

4.3와 비슷한 방식으로 User들이 평점을 매기지 않은 영화들에 대해서도 평점을 예측해 User들이 선호할 영화를 추천하고자 한다. 위의 Matrix는 Rating Matrix(671x9064)와 Item들 간의 Similarity Matrix(9064x9064)를 곱해 각 movie에 대한 user의 평점을 예측한 것이다.

4.5 User based CF와 Item based CF 결과 비교

title	rating_user	genres
Nightmare Before Christmas, The (1993)	5.0	Animation|Children|Fantasy|Musical
Clueless (1995)	5.0	Comedy|Romance
Sense and Sensibility (1995)	5.0	Drama|Romance
Like Water for Chocolate (Como agua para choco...	5.0	Drama|Fantasy|Romance
Apollo 13 (1995)	5.0	Adventure|Drama|IMAX
Circle of Friends (1995)	5.0	Drama|Romance
Legends of the Fall (1994)	5.0	Drama|Romance|War|Western
Terminator 2: Judgment Day (1991)	5.0	Action|Sci-Fi
Brady Bunch Movie, The (1995)	5.0	Comedy
Dances with Wolves (1990)	5.0	Adventure|Drama|Western

→ User의 기존 선호 영화 상위 10개

title_x	rating_expect	genres_x
Shawshank Redemption, The (1994)	236.913023	Crime|Drama
True Lies (1994)	181.068065	Action|Adventure|Comedy|Romance|Thriller
Beauty and the Beast (1991)	160.064695	Animation|Children|Fantasy|Musical|Romance|IMAX
Star Wars: Episode IV - A New Hope (1977)	142.873044	Action|Adventure|Sci-Fi
Toy Story (1995)	134.802579	Adventure|Animation|Children|Comedy|Fantasy
Ace Ventura: Pet Detective (1994)	130.915633	Comedy
Stargate (1994)	129.805754	Action|Adventure|Sci-Fi
Twelve Monkeys (a.k.a. 12 Monkeys) (1995)	128.449014	Mystery|Sci-Fi|Thriller
Fargo (1996)	128.411076	Comedy|Crime|Drama|Thriller
Pretty Woman (1990)	123.853432	Comedy|Romance

title_x	rating	genres_x
True Lies (1994)	111.133019	Action|Adventure|Comedy|Romance|Thriller
Pretty Woman (1990)	110.953000	Comedy|Romance
Beauty and the Beast (1991)	103.342063	Animation|Children|Fantasy|Musical|Romance|IMAX
Ace Ventura: Pet Detective (1994)	99.021198	Comedy
Cliffhanger (1993)	95.732183	Action|Adventure|Thriller
Dave (1993)	94.553702	Comedy|Romance
Babe (1995)	94.522159	Children|Drama
Shawshank Redemption, The (1994)	93.744924	Crime|Drama
Stargate (1994)	93.355424	Action|Adventure|Sci-Fi
Dumb & Dumber (Dumb and Dumber) (1994)	91.116827	Adventure|Comedy

→ User 기반의 rating 예측 상위 10개(왼), Item 기반의 rating 예측 상위 10개(오)

위의 테이블은 특정 User에게 User 기반의 rating 예측과 Item 기반의 rating 예측을 활용해 사용자의 평점이 가장 좋을 거 같은 영화 상위 10개를 선정해 사용자에게 추천한 결과이다. User 기반의 CF와 Item 기반의 CF의 결과를 비교하기 위해 특정 User가 기존에 선호하던 장르와 그 User에게 새롭게 추천된 영화들의 장르들을 비교하고자 한다.

Drama	6
Romance	5
Comedy	2
Adventure	2
Western	2
Fantasy	2
Musical	1
War	1
Action	1
Children	1
IMAX	1
Animation	1
Sci-Fi	1

Comedy	5
Adventure	4
Action	3
Thriller	3
Sci-Fi	3
Crime	2
Drama	2
Romance	2
Children	1
Animation	1
Fantasy	1
Mystery	1

Comedy	5
Romance	4
Adventure	4
Action	3
Children	2
Drama	2
Thriller	2
IMAX	1
Sci-Fi	1
Fantasy	1
Crime	1
Animation	1
Musical	1

→ 기존 선호 영화 장르(왼), User 기반 예측된 선호 영화 장르(중), Item 기반 예측된 선호 영화 장르(오) 목록

기존에 가장 선호하던 장르인 'Drama'와 'Romance'를 각각 비교해보면, 먼저 가장 선호됐던 'Drama' 장르에 대해서는 User 기반과 Item 기반 모두 2개의 영화만을 추천하며 기존과 비교했을 때는 다소 부족한 추천을 했다고 분석할 수 있다. 그리고 두번째로 선호됐던 Romance에 대해서는 User기반의 CF는 2개의 영화를 추천한 것에 비해 Item 기반의 CF는 4개의 영화를 추천한 것으로 보아 Item간의 유사도를 기반으로 추천한 Item based CF가 기존 사용자의 선호를 조금 더 잘 반영한 것으로 분석 가능하다.