ESC1조 데이터 분석

김민회 김수연 백채빈 손지우 이성우

INDEX

ESC

1

Continue Preprocessing

Merge 시청률 & 실적

실적 데이터에 시청률 데이터를 merge

```
# merge dataset
# average_rating
def average_rating(df,r2):
    st = str(df.start_time)  # datetime
    fi = str(df.start_time + timedelta(minutes=ceil(df['showtime'])))
    return (r2[st:fi].sum()/ceil(df['showtime']))

df['avg_rating'] = df.apply(average_rating, r2=r2, axis=1)
```

방송시간 (노출)에 따른
평균 시청률

```
# max_rating
def max_rating(df,r2):
    st=str(df.start_time)
    fi=str(df.start_time+timedelta(minutes=ceil(df['showtime'])))
    return (r2[st:fi].max())

df['max_rating']= df.apply(max_rating, r2=r2, axis=1)
```

방송시간(노출)에서의
최고 시청률

Time Slot 생성

‘주차 변수’를 추가적으로 생성

```
# timeslot : 24 * 7 = 168
df['dayofweek']     = df['broadcastTime'].dt.dayofweek      # 분(숫자)
# 0 : Mon ~ 6 : Sun
df['timeslot'] = df['dayofweek'] * 24 + df['hour'] + 1
df['timeslot'].unique()
```

```
array([ 31,  32,  33,  34,  35,  36,  37,  38,  39,  40,  41,  42,  43,
        44,  45,  46,  47,  48,  49,  50,  55,  56,  57,  58,  59,  60,
        61,  62,  63,  64,  65,  66,  67,  68,  69,  70,  71,  72,  73,
        17,  18,  19,  20,  21,  22,  23,  24,  25,  26, 123, 147,   3,
        27,  51,  75,  99])
```

	broadcastTime	showtime	product	category	price	amount	date	month	...	dayofweek	timeslot
0	2019-01-01 06:00:00	20.0	테이트 남성 셀린니트3종	의류	39900	2099000.0	2019-01-01	1	...	1	31

외부 데이터 Crawling

KOSIS

[1] 소비자동향조사(전국)

자료갱신일: 2020-07-29 / 수록기간: 월 2008.07 ~ 2020.07 / 자료문의처 : 02-759-5670

일괄설정 | 항목[1/1] | 지수코드별[37/37] | 분류코드별[28/28] | 시점[6/145]

지수코드별	분류코드별	2020. 07	2020. 06	2020. 05	2020. 04	2020. 03	2020. 02
현재생활형편CSI	전체	85	84	79	77	83	
	남자	85	84	79	76	82	
	여자	86	85	80	79	84	
	40세미만	90	90	87	87	90	
	40-50세	87	86	81	77	85	
	50-60세	82	77	74	73	79	
	60-70세	82	81	76	73	79	
	70세이상	84	84	77	71	77	
	봉급생활자	90	90	86	84	89	
	자영업자	75	69	60	57	66	
	기타	80	79	75	71	78	
	100만원미만	68	71	69	61	70	
	100-200만원	80	79	71	67	74	
	200-300만원	82	81	75	70	79	87
	300-400만원	82	79	78	78	79	87
	400-500만원	87	87	81	82	87	95
	500만원이상	94	92	88	86	92	101
	자가	84	83	78	75	82	91
	임차 등	88	86	82	81	85	93
	서울	86	85	80	78	85	93
	6대광역시	83	82	77	75	81	89
	기타도시	86	85	80	77	83	92
현재경기판단CSI	전체	49	44	36	31	38	66

Consumer Survey Index (CSI)

- 현재 생활형편
- 현재 경기판단
- 생활 형편 전망
- 향후 경기 전망
- 소비자 심리 지수

외부 데이터 Crawling

실적 데이터에 merge

	date	now_living_csi	now_judge_csi	future_living_csi	future_judge_csi	CCSI
0	2019-01	90	65	91	76	98
1	2019-02	93	70	92	80	100
2	2019-03	91	70	94	79	100
3	2019-04	93	74	95	81	102
4	2019-05	91	69	92	75	98
5	2019-06	91	69	92	75	98
6	2019-07	91	67	92	70	96
7	2019-08	90	63	89	66	92
8	2019-09	92	68	92	75	97
9	2019-10	92	72	93	77	99
10	2019-11	92	73	95	81	101
11	2019-12	92	74	94	82	101

외부 데이터 Crawling

날씨 Data

```
weather.fine_dust_grade=weather.fine_dust_grade.replace('나쁨','1')
weather.fine_dust_grade=weather.fine_dust_grade.replace('매우나쁨','1')
weather.fine_dust_grade=weather.fine_dust_grade.replace('보통','0')
weather.fine_dust_grade=weather.fine_dust_grade.replace('좋음','0')
weather.fine_dust_grade=weather.fine_dust_grade.astype('int')
weather.fine_dust_grade.unique()
```

```
array([0, 1])
```

```
weather.dtypes
```

```
place_id                    int64
weather_id         datetime64[ns]
precipitation             float64
fine_dust_grade             int64
dtype: object
```

```
weather.head()
```

	place_id	weather_id	precipitation	fine_dust_grade
0	413	2019-01-01 00:01:00	0.0	0
1	413	2019-01-01 00:02:00	0.0	0
2	413	2019-01-01 00:03:00	0.0	0
3	413	2019-01-01 00:04:00	0.0	0
4	413	2019-01-01 00:05:00	0.0	0

- 강수 유무

비가 왔다: 1, 비가 안 왔다: 0

- 미세먼지 농도

나쁨, 매우 나쁨:1, 좋음, 보통: 0

➔ 아직 PROCESSING!

Brand를 찾아서...

```
# 브랜드 변수 생성
def makeSpace(replace_list, product):
    for x in replace_list:
        product = product.replace(x, ' ')
    return product

def cleanText(readData):
    text = re.sub('[-_=,#/₩?:^$.@*₩"※~&%・!」₩] ₩|₩(₩)₩[₩]₩<₩>`₩'…》]', ' ', readData) # '+'는 그대로 놔두기
    return text

elements_to_remove = ['일','무', '뉴', '3인용', '1등급','무이자','일시불','초특가','221L','467L', '19FW','19', '20', 'FW', 'F/W', 'SS
 '세일20', 'ARS10', '실크플러스', 'by', '완벽더블', '일시불', '직매입', '특집', '여자를', '서울대', '더', '위한',
 '리얼', '뉴질랜드', '단하루', '100', '옛날', '자연산', '국내산', '특등급', '구워만든', '가격인하', '속초명물',
 '맛있는', 'A4', '제주','가','무이자', '일시불', '국내산', '김병만의', '김병지', '완벽더블구성', '기본구성', '파격가',
 '2019년', 'D', 'ALL', 'New', '1세트','2세트','5세트', '국내제작', '중형','점보특대형','점보형','퀸+퀸','킹+싱글','퀸+싱글',
 '킹사이즈','퀸사이즈','싱글사이즈', '더블+더블','더블+싱글','더블사이즈','싱글사이즈','1+1', '풀패키지','실속패키지',
 '국내제조','한세트','붙이는', 'KF94', '12', '싱글+싱글', 'NEW프리미엄', '프리미엄', '2019', 'S', '일시불', '2019년형', '1세트',
 '국내생산', '19년', '전기식', '베스트', '서장훈의','국내제조', '프랑스직수입', '신제품', '초특가']
replace_list = ['순홍삼', '19FW','19', '20', 'FW', 'F/W', 'SS']

def giveMeBrand(product):
    product = makeSpace(replace_list, product)
    product = cleanText(product)
    product = product.split()

    filtered_line = []
    for element in product:
        if element not in elements_to_remove:
            filtered_line.append(element)

    return filtered_line[0]

df['브랜드'] = df['product'].apply(lambda x: giveMeBrand(x))
```

Brand를 찾아서...

```
#예시 확인
df['브랜드']
```

```
0          테이트
1          테이트
2          테이트
3          테이트
4          테이트
          ...
37367       쿠첸
37368       쿠첸
37369       쿠첸
37370       쿠첸
37371       쿠첸
Name: 브랜드, Length: 37372, dtype: object
```

```
print(len(df['브랜드'].unique()), '여 개의 브랜드가 있다.', sep='')
```

```
430여 개의 브랜드가 있다.
```

Train Data: 브랜드 이어서 뽑아내기

- 약 430여개 브랜드
- '건강기능', '농수축'은 무의미.

Test Data: 새로운 상품 구분 필요

- 자연어 처리 공부 진행중
- 상품명 TFIDF 분석

2

Test Dataset

Mother Code

3-2. 과거실적 없는 것 찾아내기

- 19년도에는 있지만 20년도에는 없는 데이터(마더코드/상품코드 중심으로)

In [36]:
```
test = pd.read_excel('data/02_평가데이터/2020 빅콘테스트 데이터분석분야-챔피언리그_2020년 6월 판매실적예측데이터(평가데이터).xlsx', header=1)
test.head()
```

Out[36]:

	방송일시	노출(분)	마더코드	상품코드	상품명	상품군	판매단가	취급액
0	2020-06-01 06:20:00	20.0	100650	201971	잭필드 남성 반팔셔츠 4종	의류	59800	NaN
1	2020-06-01 06:40:00	20.0	100650	201971	잭필드 남성 반팔셔츠 4종	의류	59800	NaN
2	2020-06-01 07:00:00	20.0	100650	201971	잭필드 남성 반팔셔츠 4종	의류	59800	NaN
3	2020-06-01 07:20:00	20.0	100445	202278	쿠미투니카 쿨 레이시 란쥬쉐이퍼&팬티	속옷	69900	NaN
4	2020-06-01 07:40:00	20.0	100445	202278	쿠미투니카 쿨 레이시 란쥬쉐이퍼&팬티	속옷	69900	NaN

In [50]:
```
len(retail_sum['마더코드'].unique())
```

Out[50]: 711

In [51]:
```
len(test['마더코드'].unique())
```

Out[51]: 225

In [56]:
```
len(set(list(retail_sum['마더코드']) + list(test['마더코드'])))
```

Out[56]: 847

In [59]:
```
711+225-847
#89개의 마더코드가 안 겹친다.
```

Out[59]: 89

Mother Code

	방송일시	노출(분)	마더코드	상품코드	상품명	상품군	판매단가	취급액
0	2020-06-01 06:20:00	20.0	100650	201971	잭필드 남성 반팔셔츠 4종	의류	59800	NaN
1	2020-06-01 06:40:00	20.0	100650	201971	잭필드 남성 반팔셔츠 4종	의류	59800	NaN
2	2020-06-01 07:00:00	20.0	100650	201971	잭필드 남성 반팔셔츠 4종	의류	59800	NaN
3	2020-06-01 07:20:00	20.0	100445	202278	쿠미투니카 쿨 레이시 란쥬쉐이퍼&팬티	속옷	69900	NaN
4	2020-06-01 07:40:00	20.0	100445	202278	쿠미투니카 쿨 레이시 란쥬쉐이퍼&팬티	속옷	69900	NaN

```
new_mcode = []
for mcode in test['마더코드'].unique():
    if mcode not in df['mothercode'].unique():
        new_mcode.append(mcode)
print(new_mcode)
print('{}개의 새로운 마더코드가 있다.'.format(len(new_mcode)))
```

```
[100650, 100381, 100012, 100570, 100554, 100537, 100383, 100555, 100526, 100728, 100785, 100848, 100068, 100388, 100690, 100461, 100804, 100137, 100486, 100630, 100514, 100559, 100534, 100648, 100647, 100331, 100186, 100546, 100425, 100105, 100593, 100621, 100116, 100384, 100800, 100333, 100072, 100259, 100077, 100428, 100767, 100108, 100301, 100525, 100590, 100639, 100407, 100730, 100691, 100726, 100073, 100424, 100633, 100592, 100004, 100457, 100686, 100160, 100141, 100060, 100159, 100772, 100778, 100806, 100521, 100361, 100645, 100560, 100825, 100110, 100005, 100847, 100426, 100393, 100008, 100071, 100082, 100396, 100123, 100138, 100291, 100524, 100133, 100799, 100591, 100733, 100035, 100661, 100544, 100649, 100358, 100402, 100759, 100119, 100092, 100659, 100303, 100431, 100796, 100120, 100350, 100706, 100349, 100030, 100746, 100660, 100513, 100429, 100365, 100313, 100550, 100487, 100007, 100663, 100662, 100760, 100485, 100543, 100059, 100674, 100842, 100552, 100739, 100538, 100363, 100480, 100003, 100542, 100535, 100823, 100727, 100419, 100437, 100106, 100207, 100183, 100561, 100011, 100014, 100261]
140개의 새로운 마더코드가 있다.
```

New Brands 등장

이미용 - test data

```
beautyt = test.groupby('상품군').get_group('이미용')
```

```
beautyt_brand = []
elements_to_remove = ['무이자','일시불', '초특가','NEW','프리미엄']

for line in beautyt['상품명']:
    line = cleanText(line)
    line = line.split()

    filtered_line = []
    for element in line:
        if element not in elements_to_remove:
            filtered_line.append(element)
    beautyt_brand.append(filtered_line[0])



beautyt_brand = list(set(beautyt_brand))
print(beautyt_brand)
print(len(beautyt_brand), '개의 이미용 브랜드가 있다.')
```

```
['끌레드벨', '바바코코', '래쉬톡', 'VONIN', '더블모', '족선생', '제니하우스', '엘렌실라', '셀럽by재클린', '프리지아', '클린샤워', '실크테라피',
'비버리힐스폴로클럽', '에이유플러스', '바비리스', '참존', '블링썸']
17 개의 이미용 브랜드가 있다.
```

```
beautyt['브랜드'] = None
for idx, line in zip(beautyt.index, beautyt['상품명']):
    line = cleanText(line)
    line = line.split()
    for brand in beautyt_brand:
        if brand in line:
            beautyt.loc[idx, '브랜드'] = brand
```

```
beautyt[beautyt['브랜드'].isna()]['상품명']
```

```
Series([], Name: 상품명, dtype: object)
```

우선, 기존 방식 그대로 적용!
(각 상품군마다)

New Brands 등장

```
#새롭게 등장한 마더코드와 그에 해당하는 상품명
new = test[test['마더코드'].isin(new_mcode)]
new['상품명']
```

```
0          잭필드 남성  반팔셔츠 4종
1          잭필드 남성  반팔셔츠 4종
2          잭필드 남성  반팔셔츠 4종
6         바비리스 퍼펙트 볼륨스타일러
7         바비리스 퍼펙트 볼륨스타일러
                 ...
2886     쉴렉스  안마의자 렌탈서비스
2887     쉴렉스  안마의자 렌탈서비스
2888     쉴렉스  안마의자 렌탈서비스
2889      아놀드파마 티셔츠레깅스세트
2890      아놀드파마 티셔츠레깅스세트
Name: 상품명, Length: 1372, dtype: object
```

자연어 처리

1) 자카드 유사도 공식

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$

2) 코사인 유사도 공식

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

New Brands 등장

```
# 각 단어
text = tfidf.get_feature_names()

# 각 단어의 벡터 값
idf = tfidf.idf_
```

‘가구’만 먼저 시도!

```
print(dict(zip(text, idf)))
```

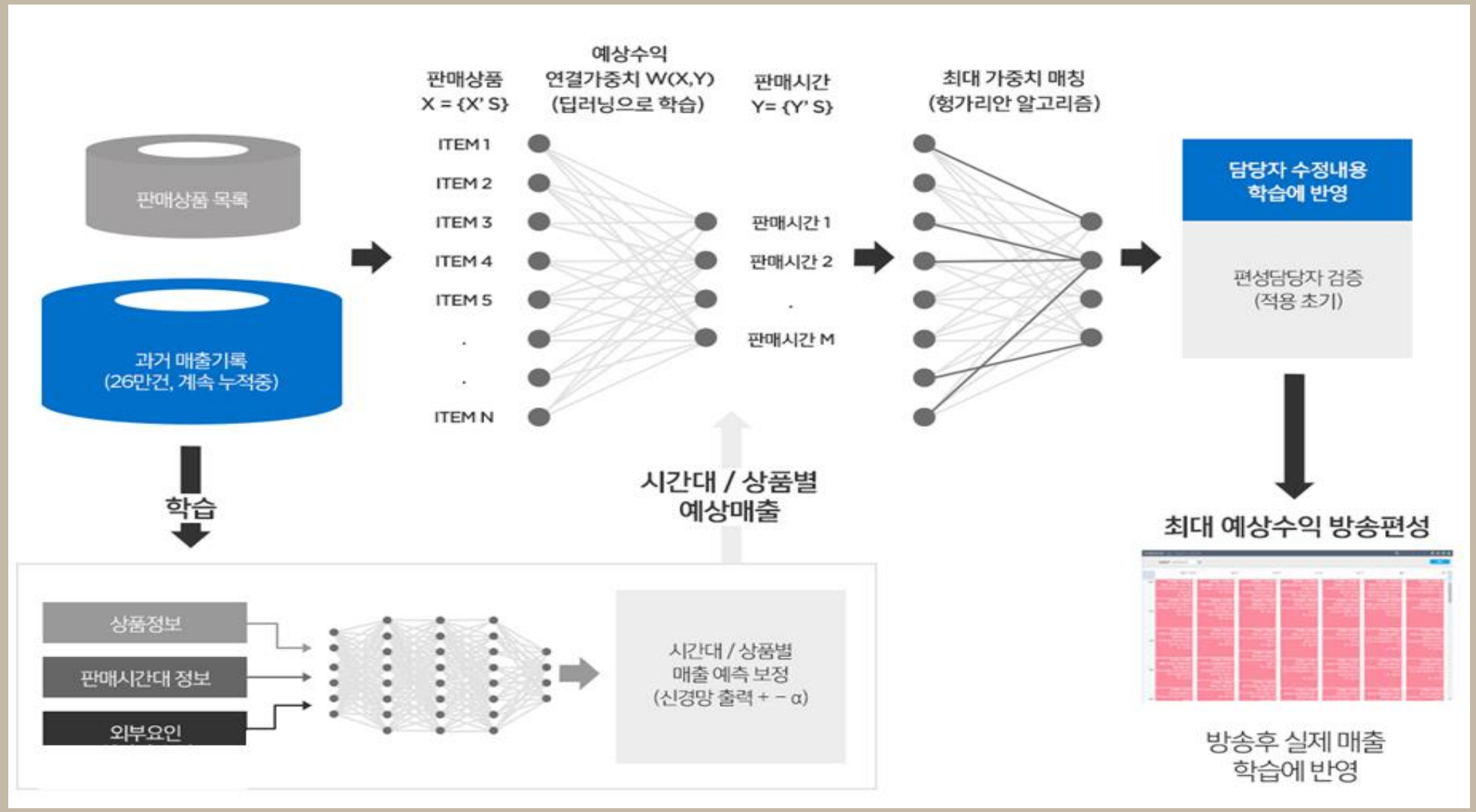
```
{'2m': 4.49650756146648, '3인용': 2.599387576580599, '5단': 4.901972669574645, '800': 4.901972669574645, 'hg': 4.901972669574645,
'led': 1.7664784536454947, 'led침대': 2.7047480922384253, 'ss': 1.9575336904082041, 'tq100': 5.189654742026425, '가죽': 5.189654742
026425, '광폭': 4.901972669574645, '기본형': 4.901972669574645, '내추럴': 4.901972669574645, '뉴퍼스티지r': 5.59511985013459, '델
라': 5.59511985013459, '레스토닉': 4.901972669574645, '루나': 2.3762440252663892, '루나시즌2': 2.7047480922384253, '리클라이너': 5.
189654742026425, '리클라이닝': 5.189654742026425, '마리노': 5.59511985013459, '매트리스': 5.59511985013459, '매트리스포함': 5.59511
985013459, '멀티': 3.649209701079277, '멀티수납형': 1.845615774204219, '모데나': 5.189654742026425, '베드룸': 5.59511985013459, '벨
라홈': 5.189654742026425, '보령황토': 5.189654742026425, '보루네오': 1.436236766774918, '붙박이장': 4.49650756146648, '블루투스':
4.901972669574645, '삼익가구': 2.599387576580599, '서랍': 4.901972669574645, '서랍장': 4.901972669574645, '서랍형': 2.992430164690206
, '세트1': 5.59511985013459, '소파': 2.5746949639902272, '수납형': 3.649209701079277, '슈퍼싱글': 1.6535380424648993, '슬라이딩':
5.189654742026425, '시공패키지': 4.901972669574645, '실버슬림': 4.901972669574645, '심플': 4.901972669574645, '싱글': 4.091042453358316, '어블러': 3.649209701079277, '원목': 5.59511985013459, '유로탑': 1.8223589120399515, '유캐슬': 5.59511985013459, '이누스바
스': 4.901972669574645, '이조농방': 5.189654742026425, '장수흙침대': 3.8903717578961645, '제니비': 2.992430164690206, '천연소가죽':
2.6506808709681495, '침대': 1.7449722484245314, '침실가구': 5.59511985013459, '퀸사이즈': 5.189654742026425, '페이지': 5.595119850134
59, '풀세트': 5.59511985013459, '프레임': 5.59511985013459, '프리미엄': 5.189654742026425, '피올레': 2.6506808709681495, '하이바
스': 4.901972669574645, '한샘': 3.8903717578961645, '협탁': 5.59511985013459, '화이트': 4.901972669574645, '화장대': 5.595119850134
59, '흙침대': 5.189654742026425}
```

```
from sklearn.metrics.pairwise import cosine_similarity
a = cosine_similarity(tfidf_matrix, tfidf_matrix)

from sklearn.metrics.pairwise import linear_kernel
cosine_sim = linear_kernel(tfidf_matrix, tfidf_matrix)

#a랑 cosine_sim이 같을듯...!
```

3

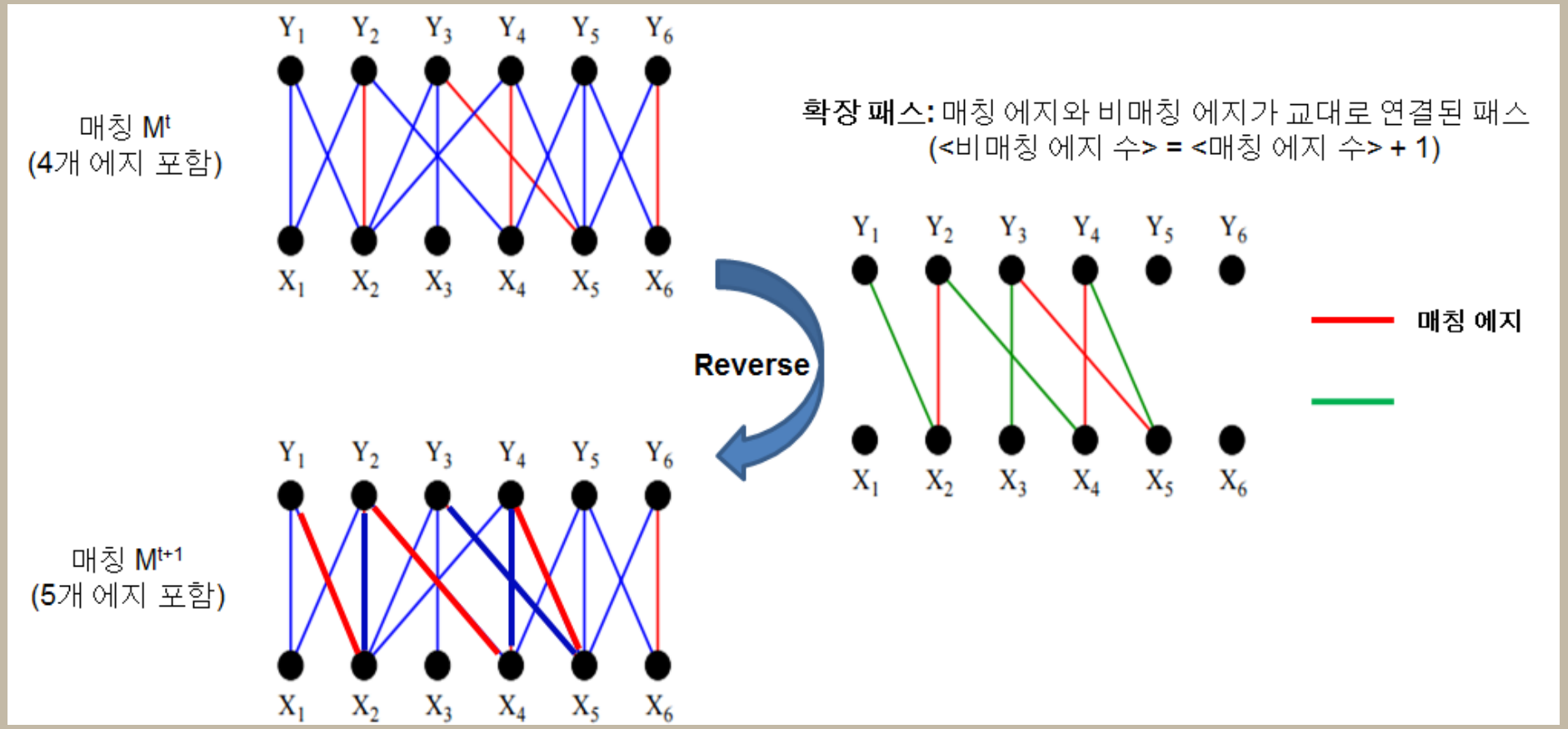
DISCUSS MODEL

Model Discussion

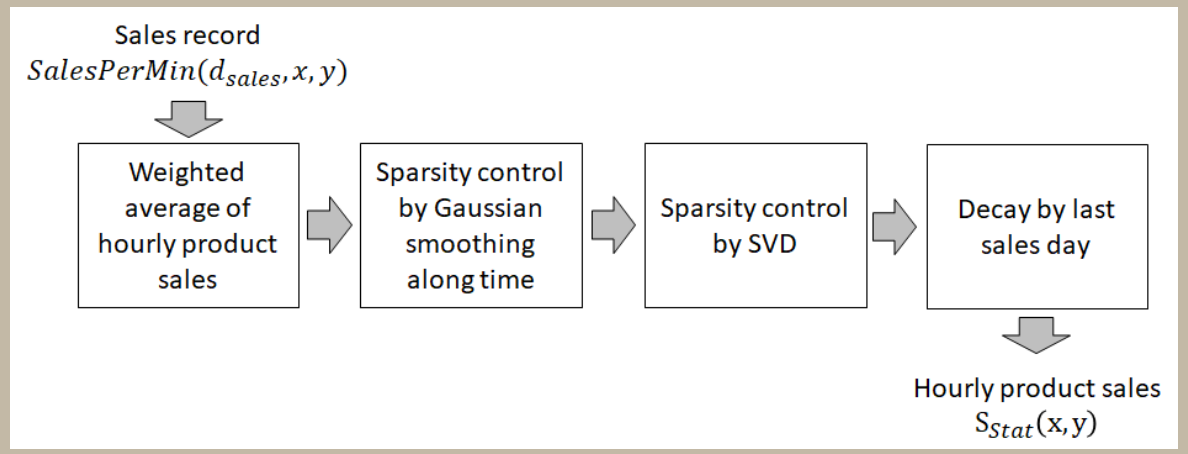


Model Discussion

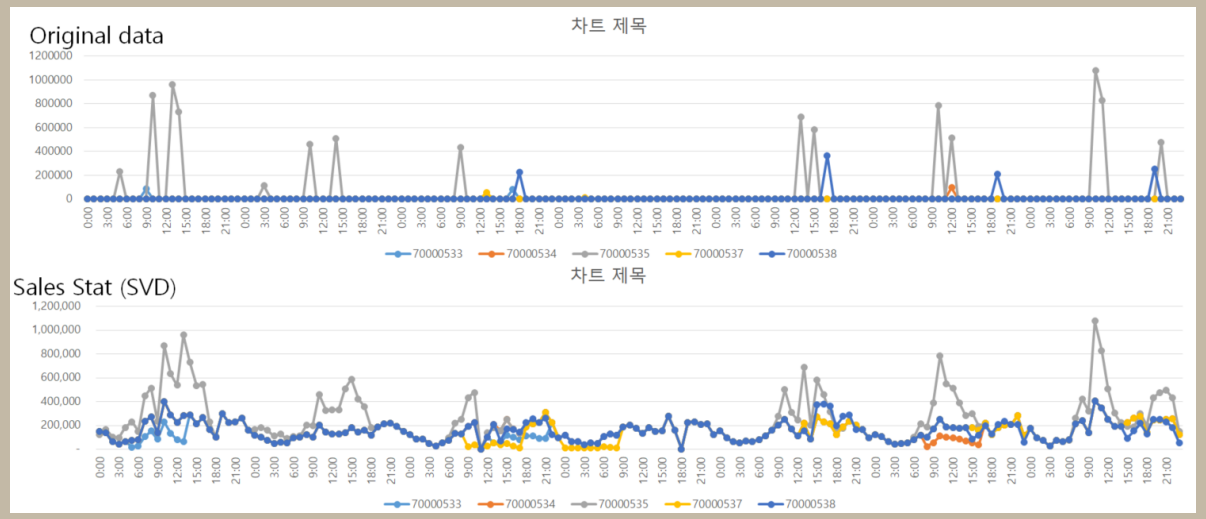
Hungarian Matching Algorithm



Model Discussion



- 시간대별 트랜드 파악
- 각 상품별 상대적 경쟁력 반영



SMOOTHING, SVD

ESC1조 데이터 분석

THANK YOU